



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

NÁVRH LED OSVĚTLENÍ PRO PRŮMYSLOVÉ
KAMEROVÉ SYSTÉMY

DESIGN OF LED LIGHTING FOR INDUSTRIAL CAMERA SYSTEMS

BAKALÁŘSKÁ PRÁCE

BACHELOR THESIS

AUTOR PRÁCE

AUTHOR

Martin Svoboda

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Ilona Janáková, Ph.D.

BRNO 2019



Bakalářská práce

bakalářský studijní obor Automatizační a měřicí technika
Ústav automatizace a měřicí techniky

Student: Martin Svoboda

ID: 177550

Ročník: 3

Akademický rok: 2018/19

NÁZEV TÉMATU:

Návrh LED osvětlení pro průmyslové kamerové systémy

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je navrhnout elektronickou, mechanickou a optickou část LED/OLED světel použitelných pro osvětlení kamerových inspekcí. Zejména by se jednalo o návrh multispektrálního osvětlení. Úkolem je také řešit řízení osvětlení, případně zpětnou komunikaci (světlo by informovalo o teplotě, intenzitě osvětlení a kolik hodin má nasvíceno) a ověření stability (měření teploty).

1. Seznamte se s problematikou návrhu osvětlení s technologií LED a používanými typy osvětlení pro strojové vidění (multispectral light, backlight, ringlight, low-angle light, domelight,...)
2. Seznamte se s technologií 3D tisku/CNC obrábění.
3. Navrhněte zvolený typ DPS s mikrokontrolérem s UDP a modulem teploměru, čidlem intenzity osvětlení a říditelný napájecí zdroj.
4. Navrhněte konstrukci krytu osvětlení a rozptylových vrstev.
5. Jednotlivě navržené části realizujte a sestavte kompletní světlo s danými parametry.
6. Vyhodnoťte (spotřeba, tepelná optimalizace, odhadovaná životnost,...).

DOPORUČENÁ LITERATURA:

GIEBL, Jan. Osvětlení ve strojovém vidění [online]. Plzeň, 2016 [cit. 2018-09-10]. Dostupné z: https://dspace5.zcu.cz/bitstream/11025/23184/1/BP_Giebl.pdf. Bakalářská práce. Západočeská univerzita v Plzni. Vedoucí práce Radek Holota.

Firemní literatura výrobců, např.: Advanced Illumination, CCS, Vision & Control, Lumimax, Chromasens, Efflux, Opto Engineering a další.

Termín zadání: 4.2.2019

Termín odevzdání: 20.5.2019

Vedoucí práce: Ing. Ilona Janáková, Ph.D.

Konzultant:

doc. Ing. Václav Jirsík, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení částí druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Abstrakt

Bakalářská práce se zabývá vývojem, návrhem a zkonstruováním multispektrálního osvětlení pro průmyslové kamerové systémy. Osvětlení je koncipováno jako modulární zařízení skládající se ze dvou částí – řídicí jednotky, se kterou je možné komunikovat primárně pomocí Modbus TCP/IP protokolu a která napájí, ovládá a sbírá informace o připojené jednotce osvětlení. Jednotka osvětlení je vybavena 64 inteligentními RGBW svítivými diodami rozmístěnými v matici 8x8 a jednou kontrolní diodou naproti níž jsou umístěny snímače intenzity osvětlení pro získání zpětné vazby o barvě a intenzitě osvětlení. Mezi svítivými diodami jsou rozmístěné dva teplotní snímače pro kontrolu teploty jednotky osvětlení. Obě jednotky jsou umístěné v robustním hliníkovém šasi. Řídicí jednotka obsahuje mikropočítač naprogramovaný v jazyce C.

Klíčová slova

Strojové vidění, multispektrální osvětlení, LED, Atmega328, BH1750, TCS34725, MCP9843 SK6812RGBW, W5500, Modbus TCP/IP, Modbus RTU, I2C, SPI, UART, Ethernet, Eagle, Fusion360, Atmel Studio

Abstract

The bachelor thesis deals with the development, design and construction of multispectral lighting for industrial camera systems. The lighting is conceived as a modular device consisting of two parts - a control unit, which can be communicated primarily through the Modbus TCP / IP protocol, which powers, controls and collects information about the connected lighting unit. The illumination unit is equipped with 64 intelligent RGBW light emitting diodes located in an 8x8 matrix and one control diode opposite to which light intensity sensors are placed to give feedback on the color and intensity of the light. There are two temperature sensors located between the LEDs to check the temperature of the lighting unit. Both units are housed in a robust aluminum chassis. The control unit contains a microcomputer programmed in C language

Keywords

Machine vision, multispectral lighting, LED, Atmega328, BH1750, TCS34725, MCP9843 SK6812RGBW, W5500, Modbus TCP / IP, Modbus RTU, I2C, SPI, UART, Ethernet, Eagle, Fusion360, Atmel Studio

SVOBODA, M. Návrh LED osvětlení pro průmyslové kamerové systémy. Brno:
Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií,
2019. 71 s. Vedoucí bakalářské práce Ing. Ilona Janáková, Ph.D..

Prohlášení

Prohlašuji, že svoji bakalářskou práci na téma „Návrh LED osvětlení pro průmyslové kamerové systémy“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 20. května 2019

.....
podpis autora

Poděkování

Děkuji vedoucí bakalářské práce Ing. Iloně Janákové, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady, konzultantu projektu v externí firmě Mgr. Romanu Procházce za odborné konzultace a rady při zpracování bakalářské práce. V neposlední řadě děkuji BcA. Magdaleně Diviškové za bezmeznou podporu a trpělivost v průběhu celého studia.

V Brně dne: 20. května 2019

.....

podpis autora

Obsah

Úvod.....	14
1. Teoretická část	15
1.1 Strojové vidění	15
1.2 Osvětlení pro strojové vidění	16
1.3 Používané typy osvětlení.....	16
1.4 Vlnová délka osvětlení	17
1.5 Geometrie osvětlení	17
1.5.1 Back Light.....	18
1.5.2 Diffuse Light.....	18
1.5.3 Bright Field	18
2. Volba komponent a testování.....	19
2.1 První vývojová sestava.....	19
2.1.1 Vývojový kit Arduino Uno	19
2.1.2 Snímač intenzity osvětlení BH1750	20
2.1.3 Snímač složek viditelného spektra světla TCS34725	21
2.1.4 Snímač teploty a tlaku MP280.....	22
3. Návrh zařízení	24
3.1 Mikropočítač AtMega328PB	24
3.1.1 Zapojení, funkce a přiřazení portů	25
3.2 Teplotní snímač MCP9843	26
3.3 Ethernet kontrolér W5500.....	26
3.3.1 Nastavení ethernet kontroléru v zařízení	28
3.3.1.1 Nastavení společných registrů	28
3.3.1.2 Nastavení registrů socketu.....	29
3.3.2 Ověření spojení	30
3.4 Komunikační sběrnice a protokoly	31
3.4.1 RS232.....	31
3.4.1.1 Seriová linka v procesoru ATmega328PB	32
3.4.2 Komunikační běrnice I2C	33
3.4.2.1 I2C sběrnice v mikropočítači ATmega328PB.....	35

3.4.2.2	Adresování zařízení na I2C sběrnici.....	35
3.4.3	Komunikační sběrnice SPI.....	36
3.4.3.1	Popis komunikace SPI.....	37
3.4.3.2	Módy časování komunikace	38
3.4.3.3	SPI v mikrokontroléru ATMEGA 328PB	39
3.4.4	Modbus	40
3.4.4.1	Funkce implementované v zařízení a jejich forma.....	41
3.4.4.2	Popis a příklad funkce 0x03	42
3.4.4.3	Popis a příklad funkce 0x06	42
3.4.4.4	Popis a příklad funkce 0x10	43
3.4.4.5	Výpočet CRC.....	43
3.4.4.6	Modbus RTU	45
3.4.4.7	Modbus TCP/IP	45
3.4.5	Komunikace se zařízením pomocí Modbus klienta.....	45
3.4.5.1	Prostředí Modbus Poll	46
3.4.5.2	Ověření komunikace Modbus RTU.....	46
3.4.5.3	Ověření komunikace Modbus TCP/IP.....	47
3.5	Konstrukce zařízení.....	48
3.5.1	Elektrická konstrukce	48
3.5.1.1	Zapojení seriové linky	48
3.5.1.2	Napájecí obvody	49
3.5.1.3	Připojení externího oscilátoru a resetovací obvod.....	50
3.5.1.4	Zapojení I2C sběrnice.....	51
3.5.1.5	Zapojení inteligentních LED diod	51
3.5.1.6	Desky plošných spojů.....	52
3.5.1.7	Jednotlivé konektory DPS a jejich zapojení.....	53
3.5.2	Mechanická konstrukce	56

3.5.2.1	Hliníkové šasi	56
3.5.2.2	Bočnice zařízení	57
3.5.2.3	Průhled pro svítivé diody.....	57
3.5.2.4	3D návrh	59
3.5.2.5	Kompletace jednotek	60
3.6	Návrh FW.....	61
3.6.1	Mapa registrů	61
3.6.2	Práce s I2C sběrnici	63
3.6.3	Práce s SPI sběrnici.....	63
3.6.4	Zpracování MODBUS	65
3.6.4.1	Zpráva MODBUS po seriové lince	65
3.6.4.2	Zpráva MODBUS po ethernetu	66
3.6.5	Ovládání inteligentních LED diod	67
4.	Závěr	69

Seznam symbolů a zkratek

LED	Light Emiting Diod – Svítivá dioda
RGBW	Red, Green, Blue, White
SRAM	Statická paměť
EEPROM	Paměť pouze ke čtení
kB	Kilobyte = 2^{10} bytů
B	Byte = 8 bitů
b	bit – dvojková číslice, nejmenší jednotka dat
USB	Univerzální sériové rozhraní
PWM	Pulsní šířková modulace
I2C	Sériová sběrnice
SPI	Sériová sběrnice
USART	Univerzální synchronní/asynchronní rozhraní
A/D	Analog / digitál
VCC	Napájecí napětí
GND	Napájecí zem
TCP	Transmission Control Protocol
IP	Internet Protocol
ALU	Aritmeticko logická jednotka
UDP	User Datagram Protocol
VDC	Napětí stejnosměrné
RJ45	Konektor pro ethernet
MAC	Fyzická adresa zařízení na síti
RST	Reset
LAN	Local Area Network
PING	Program pro test spojení v síti
Rx	Příjem
Tx	Odeslání
RS232	Rozhraní seriové komunikace
ACK	Acknowledgement - potvrzení
PDU	Protocol Data Unit
ADU	Aplication Data Unit
ID	Identifikátor
Slave ID	Identifikátor Slave zařízení
CRC	Cyklická redundantní kontrola
XOR	Logická funkce exklusivní disjunkce
DPS	Deska plošných spojů
PCB	Printed Circuit Board – ekvivalent DPS
ABS	Akrylonitril Butadien Styren
RW	Registr pro čtení i zápis
RO	Registr pouze pro čtení

Seznam obrázků

Obr. 1. Prvky strojového vidění.....	15
Obr. 2. Vlastnosti jednotlivých typů světél.....	17
Obr. 3. Vlnové délky světelného spektra [3]	17
Obr. 4. První vývojová sestava	19
Obr. 5. Rozložení pinů ATmega 328 v pouzdře DIL28	20
Obr. 6. Blokové schema BH1750[4]	21
Obr. 7. Blokové schema TCS34725[5].....	22
Obr. 8. Blokové schema BMP280[6].....	23
Obr. 9. Harvardská architektura procesoru	24
Obr. 10. Procesor ATmega328PB v 32 vývodovém pouzdře TQFP	25
Obr. 11. Přiřazení pinů a portů v použitém pouzdře.....	26
Obr. 12. Blokové schéma kontroléru W5500	27
Obr. 13 Modul s kontrolérem W5500.....	27
Obr. 14. Ověření funkčnosti spojení	31
Obr. 15. Formát zasílání zpráv na RS232	31
Obr. 16. Principiální zapojení zařízení ke sběrnici I2C	34
Obr. 17. Diagram přenosu dat na sběrnici I2C	34
Obr. 18. Běžně používané zapojení sběrnice SPI	37
Obr. 19. Časování komunikace SPI v jednotlivých módech	39
Obr. 20. Formát zprávy protokolu MODBUS	40
Obr. 21. Sekvence pro výpočet CRC pro 1 byte zprávy	44
Obr. 22. Tvar zprávy Modbus TCP	45
Obr. 23. Snímek prostředí programu Modbus Poll.....	46
Obr. 24. Nastavení komunikace v režimu Modbus RTU.....	47
Obr. 25. Přijatá data po Modbus RTU	47
Obr. 26. Nastavení komunikace v režimu Modbus TCP/IP.....	48
Obr. 27. Zapojení seriové linky	49
Obr. 28. Napájecí obvody zařízení	50
Obr. 29. Připojení resetovacího obvodu a externího oscilátoru.....	50
Obr. 30. Zapojení linek SDA a SCL komunikační sběrnice I2C.....	51
Obr. 31. Zapojení inteligentních LED diod	52

Obr. 32. Deska plošných spojů řídicí části	52
Obr. 33. Deska plošných spojů části světla	53
Obr. 34. Deska plošných spojů kontrolních snímačů	53
Obr. 35. Hliníkový profil jako základ pro šasi.....	56
Obr. 36. Jednotlivé kroky obrábění hliníkového profilu	57
Obr. 37. Porovnání vlastností Plexiglas Satinice s běžným plexisklem [15]	58
Obr. 38. Demonstrace vlastností materiálu – nízká intenzita, červená barva.....	58
Obr. 39. Demonstrace vlastností materiálu - vysoký svit, zelená barva.....	59
Obr. 40. prostředí programu Autodesk Fusion 360 s modelem jednotky osvětlení .	59
Obr. 41. Kompletace jednotky světla s osazeným průhledovým plexi a DPS snímačů umístěnou na kontrolní diodě	60
Obr. 42. Kompletace řídicí jednotky	60

Seznam tabulek

Tab. 1. Důležité parametry BH1750	21
Tab. 2. Důležité parametry TCS34725	22
Tab. 3 Důležité parametry BMP280	23
Tab. 4. Nastavení a význam bitů registru UCSR1B	33
Tab. 5 Nastavení a význam bitů registru UCSR1C	33
Tab. 6. Adresní byte snímače MCP9843	36
Tab. 7. Adresy jednotlivých snímačů na sběrnici I2C	36
Tab. 8. Nastavení rozhraní SPI1	39
Tab. 9. Funkce MODBUS	41
Tab. 10. Formát zprávy Modbus, funkce 0x03	42
Tab. 11. Formát zprávy Modbus, funkce 0x06	42
Tab. 12. Formát zprávy Modbus, funkce 0x10	43
Tab. 13. Pravdivostní tabulka XOR.....	44
Tab. 14. Napájecí konektor X1	54
Tab. 15. Konektor sériové linky X2	54
Tab. 16. Programovací konektor X3.....	54
Tab. 17. Konektor připojení části světla X4_1	54
Tab. 18. Konektor připojení Ethernet kontroléru X5.....	55
Tab. 19. Konektor připojení části snímačů X6_1	55
Tab. 20. Mapa registrů řídicí část	61
Tab. 21. Mapa registrů části světla	62
Tab. 22. Mapa příkazových registrů	62

ÚVOD

Počítačové vidění je obor zabývající se snímáním, zpracováním a vyhodnocováním digitálního obrazu. Jedná se o automatické systémy, které jsou schopny na základě obrazu a dat z obrazu získaných vyhodnocovat různé situace.

Strojové vidění je podoborem počítačového vidění, který se zaměřuje na zpracování obrazu a obrazových dat v průmyslových aplikacích. V dnešní době je strojové vidění často implementováno přímo jako jedna z částí výrobního procesu. Jedná se pak hlavně o inspekci součástí, nebo celých výrobních celků.

Ještě do nedávna nebyl aspekt osvětlení snímaného obrazu pro strojové vidění vnímán jako jedna z prioritních částí systému. Vzhledem k možnostem použití jednotlivých složek světla a jejich kombinací je ovšem za použití stejných prostředků (aplikace, kamera, prostředí, atp.) možné dosahovat různých vlastností obrazu ať už se jedná o kontrast, vyloučení určitých složek barevného spektra a dalších.

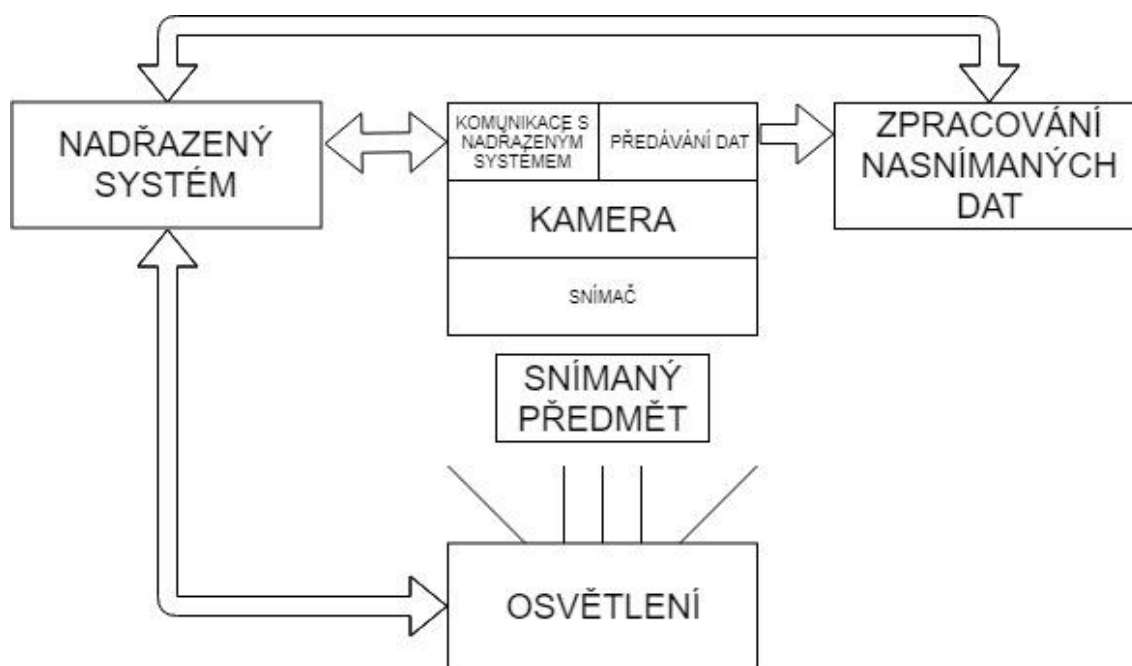
Cílem této bakalářské práce je vytvořit multispektrální osvětlení které bude odpovídat zadaným parametrům a bude použitelné pro vývojové aplikace v oblasti strojového vidění.

1. TEORETICKÁ ČÁST

1.1 Strojové vidění

Strojové vidění je podoborem počítačového vidění a sdílí s ním základní vlastnosti a principy. Tyto vlastnosti jsou popsány v následujících kapitolách

Strojové vidění je možné rozdělit na jednotlivé prvky dle následujícího schématu.



Obr. 1. Prvky strojového vidění

Kamera slouží jako prvek snímající osvětlený předmět. V dnešní době se již může jednat o inteligentní kameru, která je schopna snímání obraz zároveň vyhodnotit. V principiálním schématu v Obr. 1 je ovšem kamera, která není vybavena inteligentní elektronikou, která pouze reaguje na příkazy z nadřazeného systému a předává obrazové informace do systému který tyto informace následně zpracovává.

Zpracování nasnímaných dat je logický blok, který nasnímaná data vyhodnotí a své výsledky předá nadřazenému systému. Nadřazenému systému může dále předávat například informace o změnách ve snímaném prostředí a z toho plynoucích servisních zákroků (zaprášený senzor, nefunkční osvětlení, atd.).

Nadřazený systém synchronizuje všechny prvky strojového vidění, přijímá vyhodnocená data z kamery a podle nich upravuje řízený proces, kterého je součástí.

Osvětlení je synchronizováno a ovládáno pomocí řídicího systému. Může předávat informace o svém stavu. Primární funkcí je ovšem osvětlovat snímané prostředí podle nastavených parametrů.

1.2 Osvětlení pro strojové vidění

Pro správnou volbu osvětlení pro konkrétní účel je nutné znát parametry snímaného objektu. Zvláště pak barvu (barvy) objektu, typ a odrazivost jeho povrchu (povrchů) a rozmanitost prostorových tvarů objektu. Dle předchozích parametrů je možné zvolit konkrétní vlastnosti světla:

- Vlnovou délku
- Geometrii
- Vzor světla (maskování)
- Vhodné filtry

Pro úspěšné zvolení osvětlení a dosažení dobrých výsledků snímání obrazu a jeho následného zpracování je nutné splnit následující podmínky:

- Maximalizovat kontrast ploch jenž jsou předmětem snímání
- Minimalizovat kontrast v nezajímavých oblastech předmětu či okolí
- Zajistit stálost a opakovatelnost osvětlení

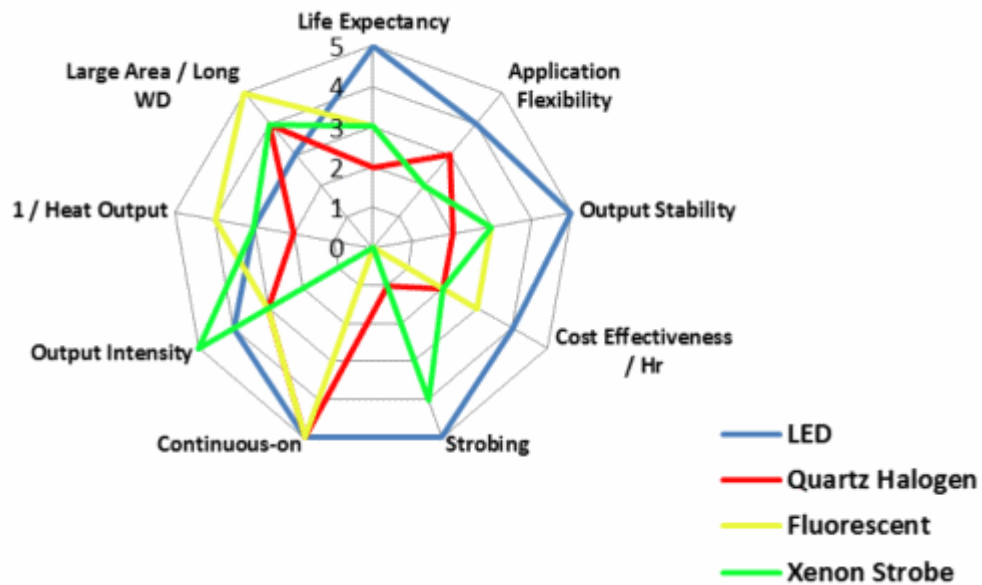
1.3 Používané typy osvětlení

Světlené zdroje pro strojové vidění užívané v současné době jsou fluorescenční, halogenové, LED, rtuťové výbojky, a xenonové výbojky.

Vlastnosti výše uvedených zdrojů světla objasňuje Obr. 2. V jednoduchosti lze říci, že pro malé plochy osvětlení se z důvodů životnosti a poměru ceny a požadovaných vlastností používají převážně LED a halogenové výbojky. Pro osvětlení větších ploch nejsou tyto zdroje dostatečné ať už svojí světelnou intenzitou, nebo cenovou dostupností. Pro osvětlení velkých ploch se používají rtuťové a xenonové výbojky.

Osvětlení hodnotíme podle následujících parametrů[1]:

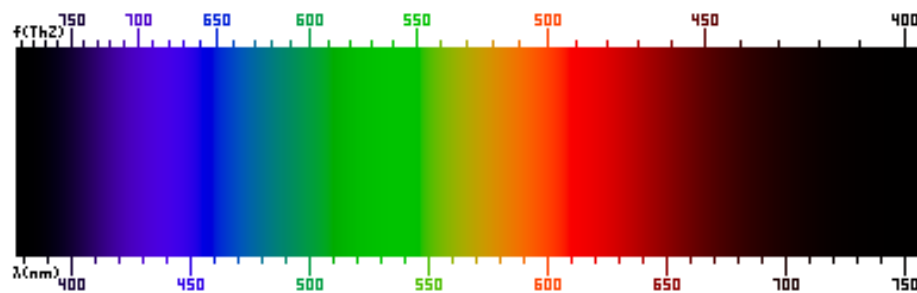
- Intenzita osvětlení (Output Intensity)
- Tepelné záření (Heat Output)
- Velikost osvětlené plochy (Large Area)
- Životnost (Life Expectancy)
- Flexibilita použití (Application Flexibility)
- Stabilita (Output Stability)
- Cenová efektivita (Cost effectiveness)
- Stroboskopické vlastnosti (Strobing)
- Nepřetržitý provoz (Continuous-on)



Obr. 2. Vlastnosti jednotlivých typů světél

1.4 Vlnová délka osvětlení

Vlnové délky světla užívané ve strojovém vidění se pohybují od 320nm (ultrafialové) až po 1000nm (infračervené).



Obr. 3. Vlnové délky světelného spektra [3]

Při zvolení vhodné vlnové délky je možné dosáhnout potlačení či zvýraznění jednotlivých oblastí snímaného objektu. Při snímání plochy s barvou konkrétní vlnové délky je vhodné pro potlačení použít tzv. komplementární barvu, neboli barvu opačnou. Při monochromatickém snímání objektu tímto postupem dosáhneme nejlepšího kontrastu snímané oblasti dané barvy.

1.5 Geometrie osvětlení

Použitím vhodné geometrie osvětlení je možné dosáhnout potlačení nežádoucích vlivů jimiž mohou být odlesky od světloodrazivých materiálů, zvlnění krycího materiálu a znemožnění snímání objektu pod kdycím materiálem. Užívané techniky svícení na snímaný objekt jsou popsány v následujících podkapitolách.

1.5.1 Back Light

Použitím Back Light, neboli zadního osvětlení dosáhneme pomocí homogenního zdroje světla prosvětlení plochy přesahující snímáný objekt. Prosvětlením plochy nezakryté snímáním objektem dostaneme kontrastní obraz objektu. Toto osvětlení je vhodné například pro zkoumání tvaru a tvarové stálosti, nebo pro detekci prasklin v objektech, kdy je pomocí zadního osvětlení prasklina ve zkoumaném objektu prosvětlena.

Pro zadní osvětlení je vhodné použít světlo s jednou dominantní vlnovou délkou, čímž je zajištěn dobrý kontrast

1.5.2 Diffuse Light

Diffuse Light, neboli rozptýlené světlo je vhodné pro snímání zvlňných, či lesklých ploch, od kterých by se při přímém osvětlení světlo odráželo a nebylo by tak dosaženo požadovaného výsledku.

Principiálně se světlo rozptýlí podle předem daného zvlňení tak, aby se odráželo od konkrétního místa. V mnoha případech je před kamerou štěrba, která propustí jen paprsky z určitého směru.

1.5.3 Bright Field

Bright Field, neboli jasné pole je pak pomyslným opakem rozptýleného světla, kdy je osvětleno, většinou bodovým zdrojem světla, právě výše uvedené zvlňení povrchu, které ale nyní není na závadu, naopak je nutné ho detekovat, protože může jít o vadu povrchu snímaného objektu.

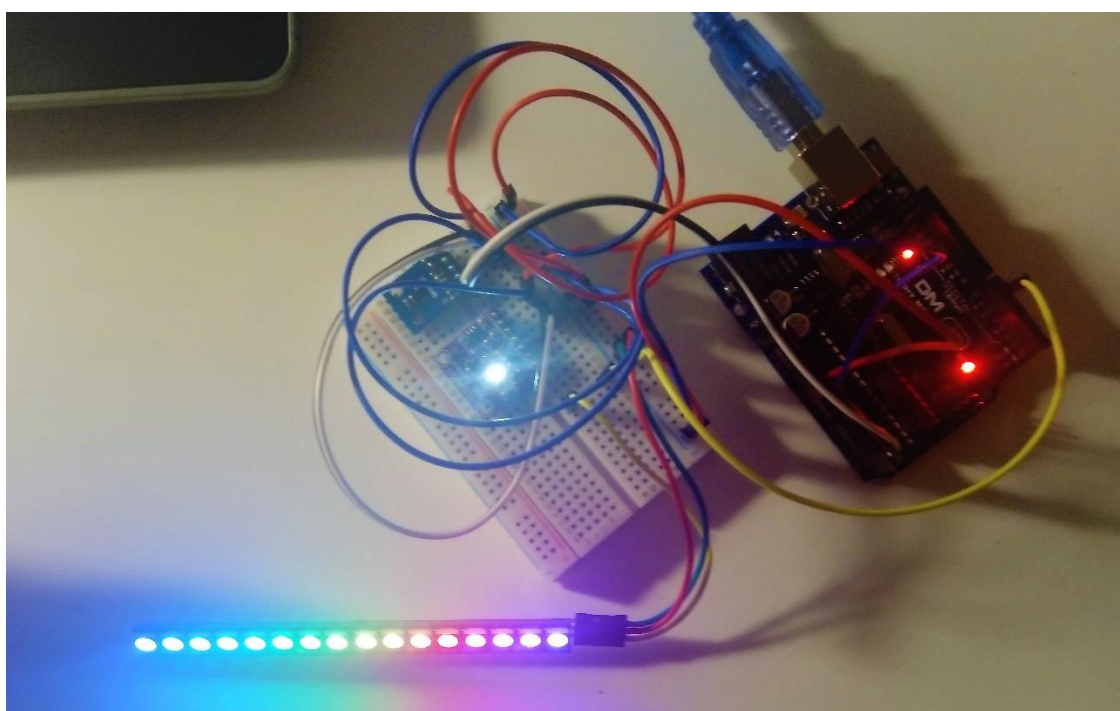
2. VOLBA KOMPONENT A TESTOVÁNÍ

2.1 První vývojová sestava

Pro otestování zvolených komponent a navrhovaného principu řízení jednotlivých LED a datového propojení se snímači byla použita vývojová sestava blíže popsaná v této kapitole.

Pro sestavu bylo použito vývojového kitu Arduino Uno, který poskytuje jak procesor s externím oscilátorem (je potřeba pro komunikaci s LED), tak i uživatelsky přívětivé prostředí, ve kterém lze jednoduše programovat.

Jako zdroj světla byly zvoleny inteligentní RGBW diody, které je možné spojit v sérii a při dostatečně silném napájecím zdroj tak mohou vytvořit v podstatě neomezeně velkou řízenou světelnou plochu. Komunikace s diodami probíhá pouze po jednom vodiči, další 2 vodiče zajišťují napájení – toto trojvodičové připojení ušetří místo na desce plošných spojů.



Obr. 4. První vývojová sestava

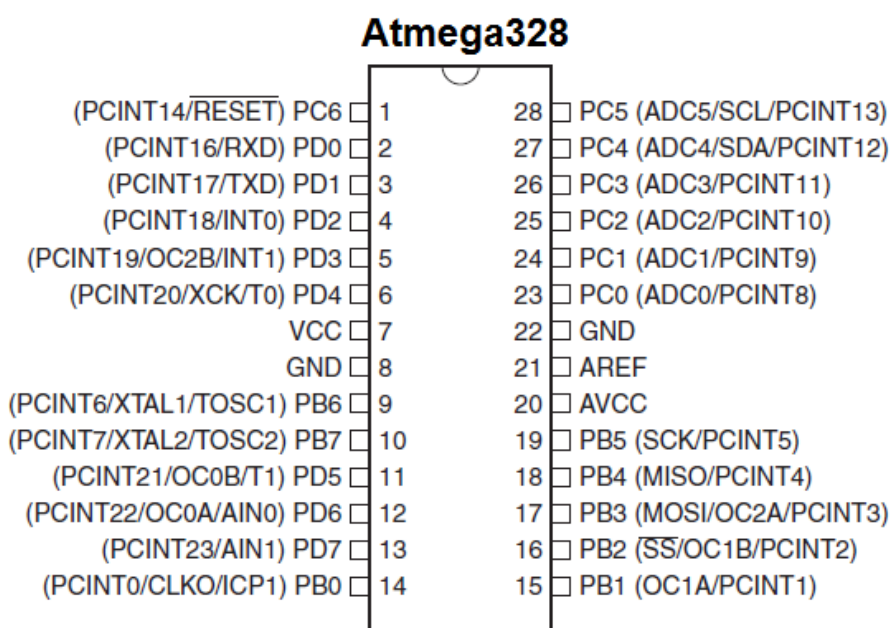
2.1.1 Vývojový kit Arduino Uno

Arduino Uno je open source vývojový kit s mikropočítač ATmega 328 s 2kB SRAM paměti a 1kB EEPROM paměti, taktovaný 16MHz externím oscilátorem. Kit zpřístupňuje všechny potřebné vstupy a výstupy mikročipu a jeho komunikační

rozhraní vyvedené na dutinkové lišty po stranách desky plošného spoje. Vývojový kit umožňuje zavádět do mikročipu program bez dalších prostředků pouze pomocí připojení USB kabelem k počítači díky zavedenému bootloaderu.

Mikropočítač ATmega328 disponuje následujícími periferiemi:

- 14 digitálních vstupů/výstupů
- 6 analogových vstupů/výstupů
- 6 PWM výstupů
- Komunikační rozhraní I2C, SPI a UART



Obr. 5. Rozložení pinů ATmega 328 v pouzdře DIL28

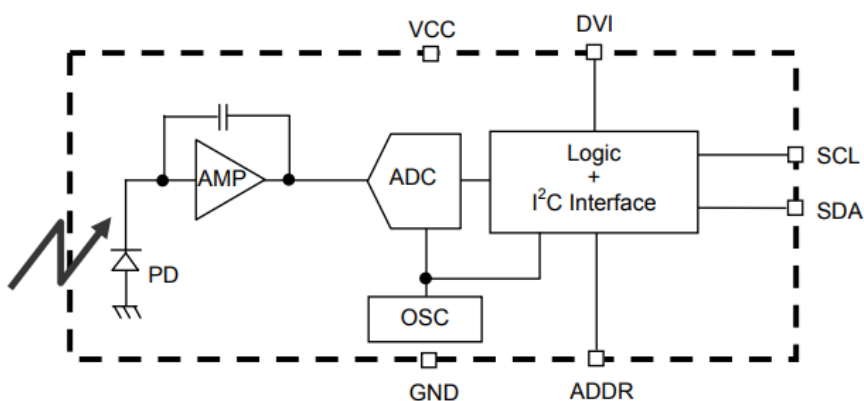
2.1.2 Snímač intenzity osvětlení BH1750

Snímač intenzity osvětlení BH1750 je snímač světelnou intenzitu a převádí ji na digitální hodnotu v jednotkách Lux. Snímač je vybaven polem fotodiod, která při osvětlení generují napětí odpovídající intenzitě osvětlení – toto napětí je pak přes 16-bitový A/D převodník převedeno na digitální hodnotu poskytovanou přes I2C sběrnici nadřazenému mikropočítači. Snímač poskytuje hodnotu intenzity osvětlení přímo v jednotkách lux – není tedy nutný následný převod.

Tab. 1. Důležité parametry BH1750

Parametr	Symbol	Hodnota	Jednotka
Napájecí napětí	V _{cc}	2,4~3,6	V
Provozní teplota	T _{oper}	-40~85	°C
Maximální intenzita snímaného osvětlení* ¹	Lint _{max}	1	lx
Minimální intenzita snímaného osvětlení* ¹	Lint _{min}	65535	lx
Rozlišení * ¹	R	1	lx
Doba měření * ¹	t _{meas}	120	ms
Snímaná vlnová délka světla	λ	450~680	nm

*1 – pro použitý pracovní režim snímače (H-Resolution)



Obr. 6. Blokové schéma BH1750[4]

Snímač poskytuje zpětnou vazbu o intenzitě světla v dostatečném rozlišení pro informativní účely. Ve finálním zařízení bude snímač umístěn na kontrolní diodě světla.

2.1.3 Snímač složek viditelného spektra světla TCS34725

Snímač jednotlivých složek světla (RGBC – červená, zelená, modrá a nefiltrované čisté světlo). Snímač je vybaven polem fotodiod (3x4), před kterým je umístěn filtr blokující infračervené záření. Jednotlivé trojice fotodiod pak před sebou mají umístěny filtry propouštějící jednotlivé složky světla (R,G,B), poslední trojice je pak bez filtru. Jednotlivé trojice fotodiod při osvětlení generují napětí odpovídající intenzitě osvětlení snímané složky světla – toto napětí je pak přes 16-bitový A/D převodník (jeden převodník pro každou trojici) převedeno na digitální hodnotu poskytovanou přes I2C sběrnici nadřazenému MCU.

Snímač poskytuje hodnotu naintegrované intenzity světla za určitý čas. Tento čas je nutné podle požadovaných hodnot spočítat dle následující rovnice.

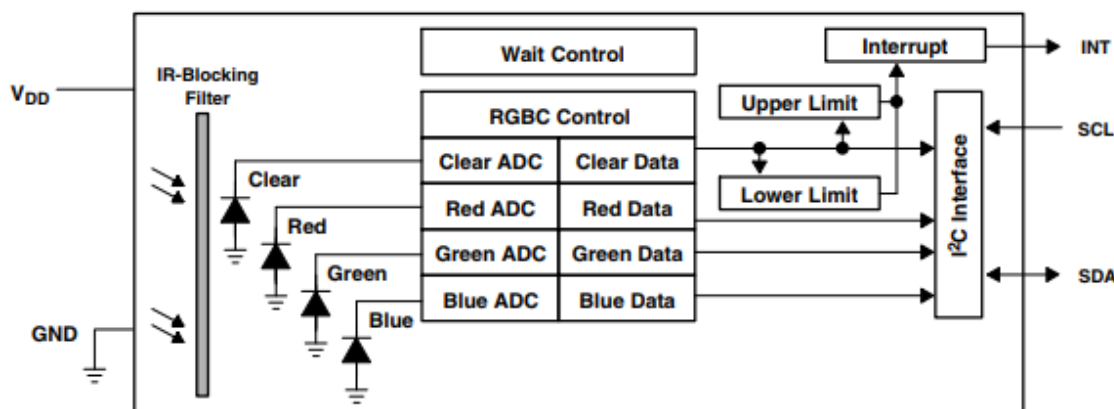
$$ATIME = 256 - \frac{\text{integrační čas}[ms]}{2,4[ms]}$$

Kde 2,4 ms je délka integračního kroku, která je pevně daná. Pro integrační čas 100ms dostáváme hodnotu 214, kterou je následně zapsána do příslušného registru snímače a zpřístupněna pomocí sběrnice I2C nadřazenému mikropočítači.

Naměřené hodnoty jsou získávány přečtením jednotlivých registrů, ve kterých jsou data po každém měření uložena.

Tab. 2. Důležité parametry TCS34725

Parametr	Symbol	Hodnota	Jednotka
Napájecí napětí	Vcc	2,7~3,6	V
Provozní teplota	Toper	-40~85	°C
Snímaná vlnová délka světla	λ	400~650	nm



Obr. 7. Blokové schema TCS34725[5]

2.1.4 Snímač teploty a tlaku MP280

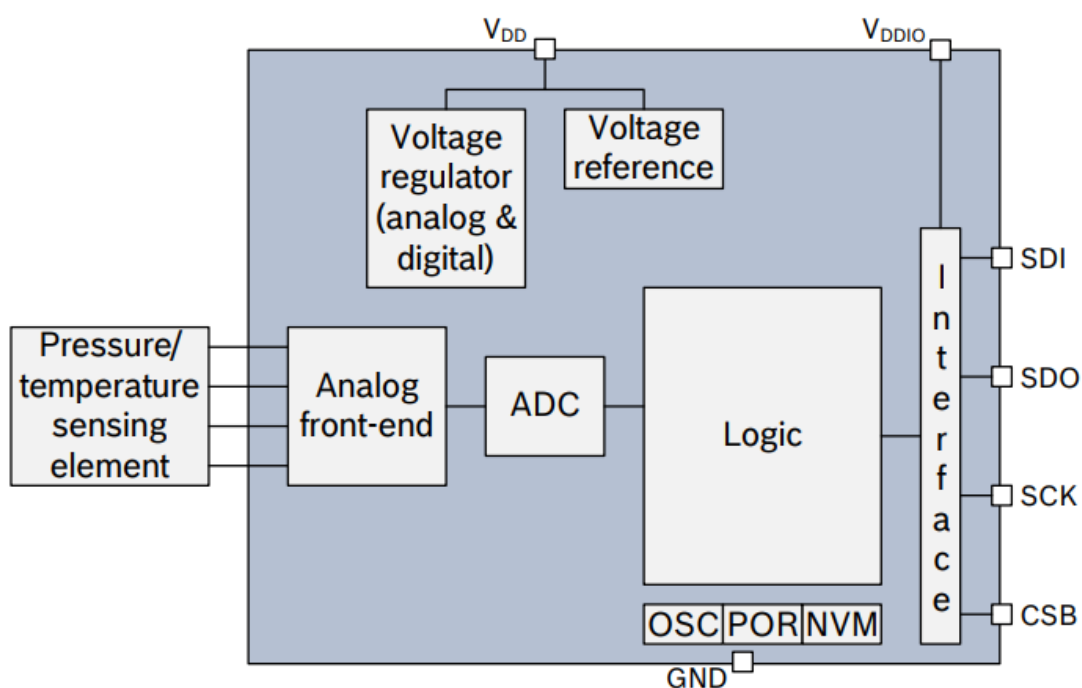
Snímač tlaku byl použit ve vývojové sestavě, pro jeho malý rozsah měřené teploty a měření teploty okolí byl zvolen jiný teplotní snímač popsany v dalších kapitolách

Snímač tlaku a teploty komunikující na sběrnici I2C. Dle nastavení snímače je možné využít až 20 bitového rozlišení. Snímač poskytuje surové hodnoty, které je následně nutné přepočítat.

Tab. 3 Důležité parametry BMP280

Parametr	Symbol	Hodnota	Jednotka
Napájecí napětí	Vcc	2,7~3,6	V
Provozní teplota	Toper	-40~85	°C
Rozsah měřené teploty	Tmp	-40~85	°C
Rozlišení	r	0,01	°C
Přesnost měření teploty	P	±1 (±0,5*2)	

*2 – v okolí teploty 25°C.



Obr. 8. Blokové schéma BMP280[6]

3. NÁVRH ZAŘÍZENÍ

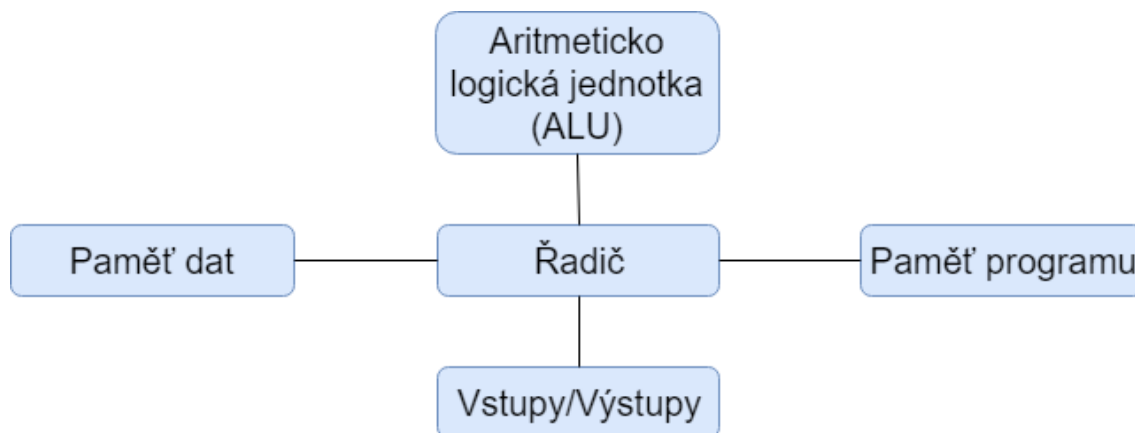
Zařízení je koncipované jako modulární, rozdělené na 2 části – řídicí jednotku, u které se předpokládá její možné použití s více typy jednotek osvětlení.

- Řídicí jednotka
 - zajišťuje komunikaci s nadřazeným systémem pomocí ethernet modulu protokolem Modbus TCP/IP, zpracování dat ze snímačů, napájení, ovládání a sběr dat ze snímačů umístěných v jednotce osvětlení.
- Jednotka osvětlení
 - obsahuje pole inteligentních diod o velikosti 8x8, dále jednu kontrolní diodu, na proti které jsou umístěny kontrolní snímače světla (TCS34725 a BH1650) odstíněné od okolního osvětlení. Pro informace o teplotě jsou na desce plošného spoje umístěny 2 teplotní snímače

V následujících podkapitolách jsou podrobně popsány komponenty, které nebyly součástí vývojové sestavy (kapitoly 2), komunikační protokoly a sběrnice, které jsou v zařízení využívány a jejich použití, konstrukce návrhu zařízení rozdělená na elektrickou a mechanickou část.

3.1 Mikropočítač AtMega328PB

ATMEGA328PB je 8-bitový mikropočítač typu RISC (redukovaná instrukční sada se 131 instrukcemi), Harvardské architektury. Harvardskou architekturou procesoru je určeno, že paměť dat a paměť program jsou od sebe oddělené.



Obr. 9. Harvardská architektura procesoru

Mikropočítač obsahuje 32kB ISP flash paměť pro program, 1kB EEPROM paměť pro uchování dat i po odpojení napájení, 2kB SRAM paměť. Z komunikačních rozhraní - 2 linky UART, 2 linky SPI, a 2 linky I2C. 27 Vstupně-výstupních programovatelných linek. Dále 2 osmi a 3 šestnácti bitové čítače/časovače a 8 ADC převodníků

Napájecí napětí procesoru je 1.8 – 5.5V (pro použití s 16MHz externím oscilátorem 4.5 – 5.5V)



Obr. 10. Procesor ATmega328PB v 32 vývodovém pouzdře TQFP

3.1.1 Zapojení, funkce a přiřazení portů

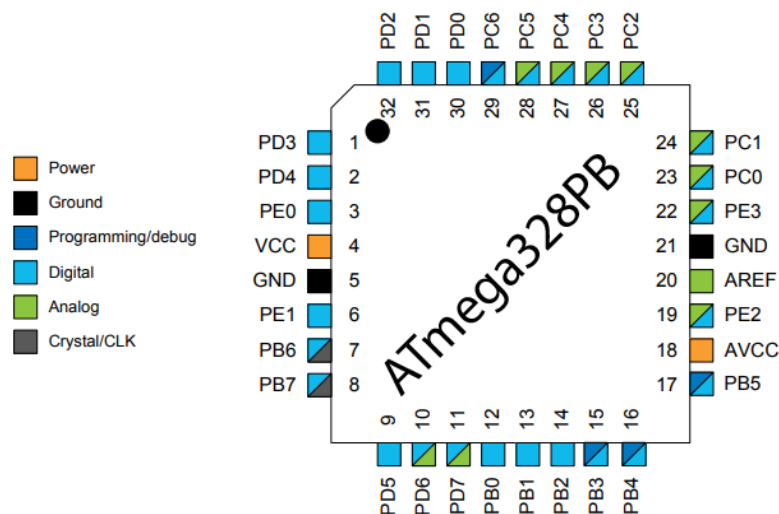
Část portu B je využita pro komunikaci na seriové lince (piny 3 a 4) a zároveň pro vyvedení programovacích pinů (piny 3, 4 a 5). K pinům 6 a 7 je pak připojen externí oscilátor. Zbytek portu B je nevyužit.

Port C je z části využit pro komunikaci po sběrnici SPI (piny 0 a 1), z části pro komunikaci po sběrnici I2C (piny 4 a 5). Pin 6 je pak vyhrazen pro reset. Zbytek portu C je nevyužit.

K portu D je na pinu 5 připojen SS vodič pro komunikaci po sběrnici SPI, pin 4 je vývodem pro komunikaci s inteligentními diodami a porty 5 a 6 jsou vyvedeny na konektor jako rezerva. Zbytek portu D je nevyužit.

Z portu E je využitý pouze pin 3, ke kterému je připojen komunikační vodič pro SPI.

Podrobné informace o zapojení jsou uvedené v kapitole 3.5.1 – Elektrická konstrukce



Obr. 11. Přiřazení pinů a portů v použitém pouzdře

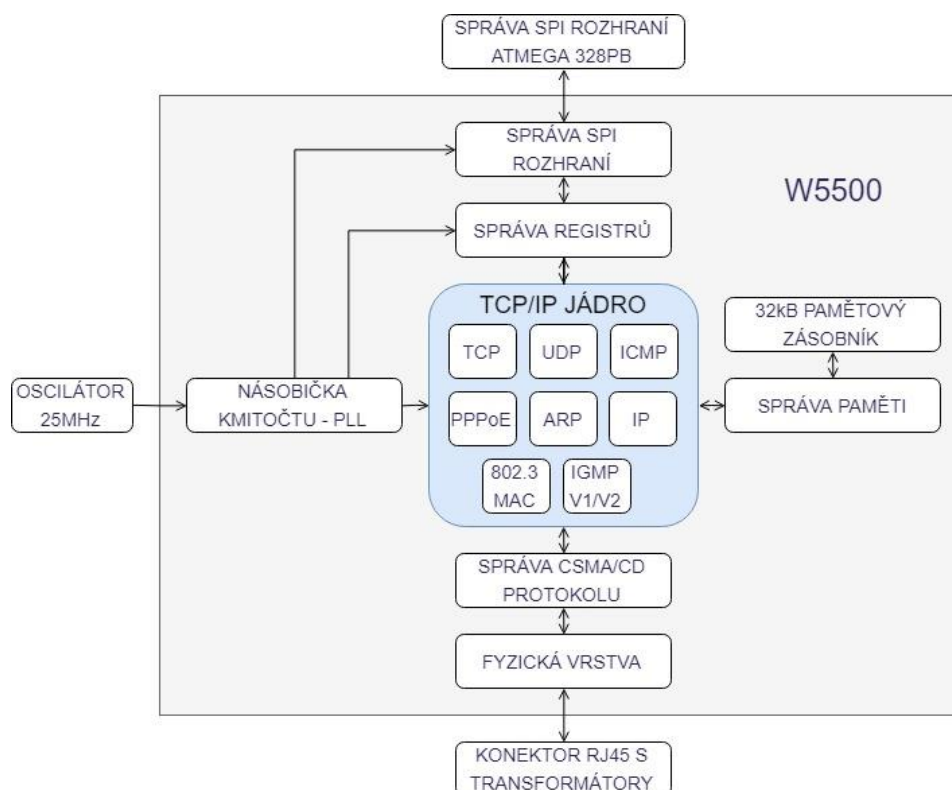
3.2 Teplotní snímač MCP9843

Teplotní snímač z vývojové sestavy neodpovídal svým rozsahem měřené teploty a způsobem měření teploty (měření okolní teploty) účelu, proto byl zvolen snímač MCP9843 s rozsahem měřených teplot -20°C až $+125^{\circ}\text{C}$, který snímá teplotu téměř kontaktně s deskou plošných spojů. Snímač pracuje s nastaveným rozlišením 0.25°C , což je pro zpětnou vazbu o teplotě jednotky osvětlení dostačující, s přesností $\pm 0,2^{\circ}\text{C}$ v oblasti $+75^{\circ}\text{C}$ až $+95^{\circ}\text{C}$, $\pm 0.5^{\circ}\text{C}$ v oblasti $+40^{\circ}\text{C}$ až $+125^{\circ}\text{C}$ a $\pm 1^{\circ}\text{C}$ v oblasti -20°C až $+125^{\circ}\text{C}$. Hodnota měřené teploty je předávána na sběrnici I2C jako 12bitové číslo kódované v dvojkovém doplňku rozdělenou do dvou bytů, které jsou doplněné o další informace (např. informace o přesahující hodnotě měřené teploty vůči nastavenému limitu, atd). Tuto hodnotu je tedy nutné následně přepočíst. Měřená teplota je pak předávána nadřazenému systému ve 2 bytech, z nichž vrchní byte obsahuje celočíselnou teplotu a horní bajt část hodnoty teploty za desetinnou čárkou.

3.3 Ethernet kontrolér W5500

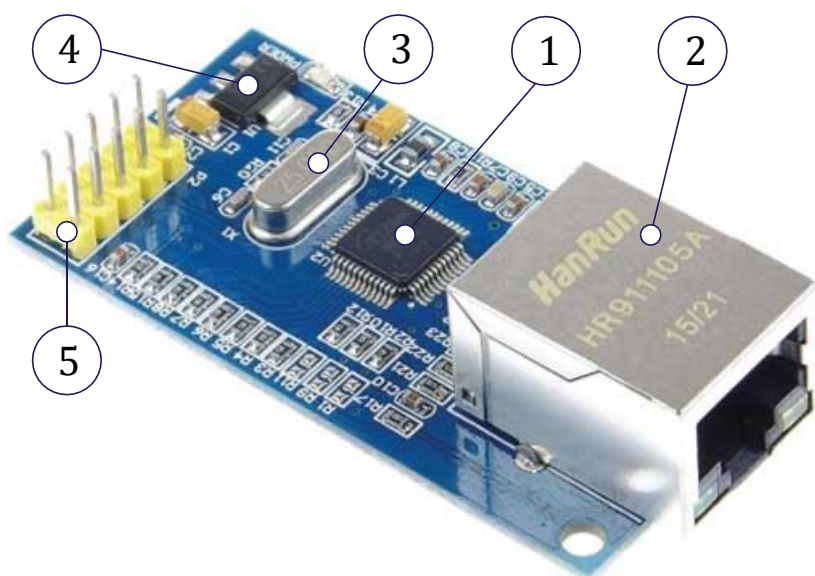
Obvod W5500 zprostředkovává komunikaci na síti ethernet. Podporuje protokoly TCP, UDP, IPv4, ICMP, ARP, IGMP a PPPoE. Obvod je vybaven osmi nezávislými sockety, kdy je možné každý socket ovládat samostatně a implementovat tak více protokolů. Přijímací a odesílací zásobníky dat mají dohromady velikost 32kB. Na fyzické vrstvě podporuje standardy 10BaseT a 100BaseTX.

Obousměrná komunikace s procesorem probíhá na sběrnici SPI rychlostí až 80Mhz. Komunikace na sběrnici SPI může probíhat v režimech „0“ a „3“.



Obr. 12. Blokové schéma kontroléru W5500

Fyzická implementace kontroléru v řídicí jednotce je realizovaná pomocí zakoupeného modulu osazeného kontrolérem (Obr. 13-1), konektorem RJ45 s integrovanými transformátory (Obr. 13-2), stabilizátorem napětí 3,3VDC (Obr. 13-4), 25MHz oscilátorem (Obr. 13-3), konektorem 2x5 kolíků s roztečí 2,54mm (Obr. 13-5) a veškerou potřebnou součástkovou obestavbou. Kontrolér je takto možné zapojit do připraveného konektoru a bez dalších úprav používat.



Obr. 13 Modul s kontrolérem W5500

Modul je připojen k řídicí desce pomocí protikusu kolíkového konektoru. Zapojení a popis jednotlivých kolíků konektoru je popsán v kapitole 3.5.1.7. Blokové schéma připojení kontroleru je popsáno v Obr. 12.

3.3.1 Nastavení ethernet kontroléru v zařízení

Pro použití ethernet kontroléru v zařízení pro přenos dat protokolem Modbus TCP/IP je nutné jeden z jeho socketů nastavit.

Registry kontroléru jsou rozdělené do 2 částí. Společná část (Common Registers) a socketová část (Socket Registers) pro každý z osmi socketů. Ve společné části jsou definovány základní informace jako MAC adresa, maska podsítě a další, které jsou pro kontrolér, potažmo všechny sockety, identické. V socketové části se vždy pro konkrétní socket nastavuje podrobná konfigurace a registry obsahují informace o přijatých a odeslaných datech.

V následujících podkapitolách jsou popsány nastavení jednotlivých registrů pro zařízení.

3.3.1.1 Nastavení společných registrů

Registr **Mode (MR)**

- Osmibitový registr z jehož jednotlivých bitů je nastavován pouze bit RST, při jehož nastavení na hodnotu „1“ jsou před inicializací vymazány veškeré registry.
- Registr dále obsahuje například nastavení funkce probuzení po LAN, zablokování odpovědi na PING, apod.

Registr **Gateway Address (GAR)**

- 4 osmibitové registry, do kterých je ukládána adresa brány sítě.
- Tento registr je v zařízení možné nastavit a tím změnit adresu brány sítě, do které je zařízení připojeno.
- Výchozí hodnota registrů je rovna [192][168][0][1], což odpovídá adrese 192.168.0.1

Registr **Subnet Mask (SUB)**

- 4 osmibitové registry, do kterých je ukládána adresa masky podsítě.
- Tento registr je v zařízení možné nastavit a tím změnit adresu masky podsítě, do které je zařízení připojeno.
- Výchozí hodnota registrů je rovna [255][255][255][0], což odpovídá adrese 255.255.255.0

Registr [Source Hardware Address \(SHAR\)](#)

- 6 osmibitových registrů, do kterých je ukládána fyzická adresa zařízení, běžně nazývaná MAC.
- Tento registr je v zařízení možné nastavit a tím změnit MAC adresu zařízení.
- Výchozí hodnota registrů je rovna [1][2][3][4][5][6], což odpovídá adrese 1:2:3:4:5:6.

Registr [Source IP Address \(SIPR\)](#)

- 4 osmibitové registry, do kterých je ukládána IP adresa zařízení
- Tento registr je v zařízení možné nastavit a tím změnit IP adresu zařízení.
- Výchozí hodnota registrů je rovna [192][168][0][2], což odpovídá IP adrese 192.168.0.2

3.3.1.2 Nastavení registrů socketu

Z osmi dostupných socketů kontroléru je v zařízení použit pouze jeden – Socket 0.

Registr [Mode \(S0_MR\)](#)

- Osmibitový registr, pomocí kterého se nastavuje mód, ve kterém je konkrétní socket používán.
- Spodní 4 bity registru určují použitý protokol. Zařízení je nastaveno v módu TCP, spodní 4 bity tedy obsahují hodnotu 1 (bitově 0001).
- Zbylé 4 bity registru nejsou nastaveny, nesouvisí totiž přímo s použitým TCP protokolem

Registr [Socket Source Port \(S0_PORT\)](#)

- 2 osmibitové registry, do kterých je ukládán komunikační port zařízení. Vzhledem k použití protokolu Modbus TCP/IP, který číslo portu definuje, je tento port nastaven na hodnotu 502 a je neměnný.

Registr [Destination IP Address \(S0_DIPR\)](#)

- 4 osmibitové registry, do kterých je ukládána IP adresa cílového zařízení.
- Tento registr je v zařízení možné nastavit a tím změnit IP adresu zařízení.
- Výchozí hodnota registrů je rovna [192][168][0][6], což odpovídá IP adrese 192.168.0.6

Registr **Destination Port (S0_DPORT)**

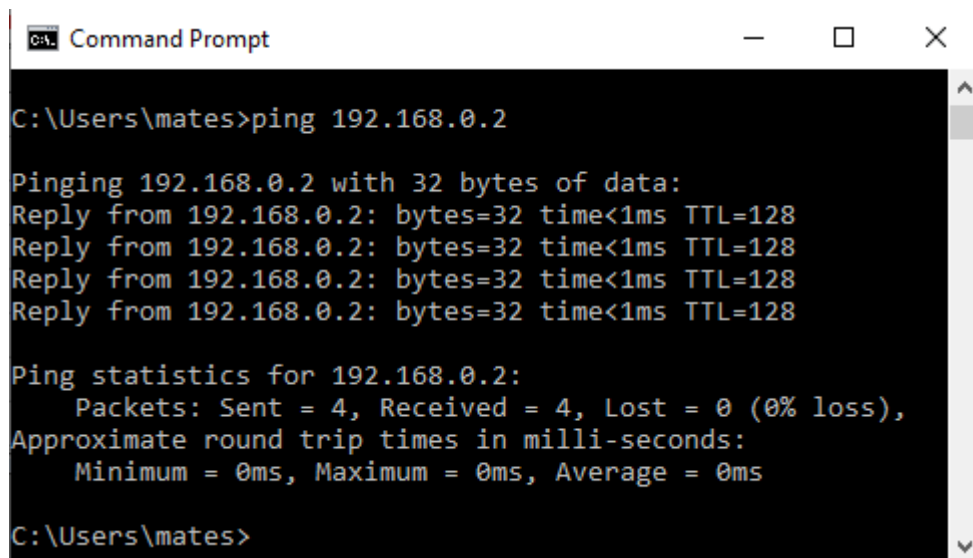
- 2 osmibitové registry, do kterých je ukládán komunikační port cíle. Vzhledem k použití protokolu Modbus TCP/IP, který číslo portu definuje, je tento port nastaven na hodnotu 502 a je neměnný.

Registr **Socket Command (S0_CR)**

- Osmibitový registr, kterým jsou předávány příkazy konkrétnímu socketu
 - Zápisem hodnoty 0x01 do registru je socket inicializován dle nastavení předchozích registrů
 - Zápisem hodnoty 0x02 je socket přepnut do módu poslechu a funguje jako TCP server. Zjednodušeně čeká, až se k němu připojí klient.
 - Zápisem hodnoty 0x04 je socket přenut do módu TCP klienta a snaží se připojit k definovanému TCP serveru. Tento mód v zařízení není použit, zařízení je konfigurováno jako TCP server.
 - Zápisem hodnoty 0x08 do registru dojde k ukončení spojení. Tento příkaz je nezávislý na konfiguraci zařízení – k odpojení dojde jak v módu TCP klient, tak v módu TCP server.
 - Zápisem hodnoty 0x10 do registru dojde k uzavření portu
 - Zápisem hodnoty 0x20 do registru dojde k odeslání dat připravených v zásobníku dat. Před zápisem tohoto příkazu je nutné připravit do zásobníku data, a do příslušných registrů sockety zapsat jejich polohu (adresu od které data začínají) a jejich velikost (počet uložených bajtů). Pokud tyto registry nejsou vyplněny, kontrolér odesílá na síť celý obsah zásobníku (32kB dat).
 - Zápisem hodnoty 0x40 do registru je kontroléru sdělena informace o přijetí dat připojeným mikrokontrolérem. Před zápisem tohoto příkazu je nutné do příslušných registrů zapsat polohu (adresu) dat, které mikrokontrolér přijal.

3.3.2 Ověření spojení

Po nastavení všech registrů uvedených výše je zařízení připraveno ke komunikaci v síti. To je možné ověřit pomocí programu Ping.exe. Program ping odesílá IP datagramy na určenou adresu a na každý datagram pak očekává odpověď. Zároveň měří časovou odezvu, respektive délku mezi odesláním datagramu a příjmem odpovědi od zařízení.



```
C:\Users\mates>ping 192.168.0.2

Pinging 192.168.0.2 with 32 bytes of data:
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\mates>
```

Obr. 14. Ověření funkčnosti spojení

Z výše uvedeného snímku programu je patrné, že po odeslání 4 datagramů do ethernet kontroléru připojeného do sítě, kontrolér navrátil všechny 4 datagramy nazpět, s minimálním zpožděním. Spojení je tedy možné považovat za funkční.

3.4 Komunikační sběrnice a protokoly

3.4.1 RS232

RS232 je asynchronní sériová obousměrná komunikační linka využívající bitového datového přenosu. Data jsou tedy bit po bitu přenášena na lince za sebou. K přenosu dat se využívají 2 vodiče. Z pohledu jednoho z komunikujících zařízení se vždy jedná o vodič Rx, na kterém zařízení data přijímá, a vodič Tx, po kterém zařízení data odesílá. Tím, že je pro přenos dat v jednom směru vždy využit samostatný vodič je na fyzické vrstvě komunikace bezkolizní.

Formát zasílání zpráv je následující: První odešle zařízení na linku StartBit, za ním následují vysílaná data (typicky 8 bitů) tak, že první odchází LSB (nejméně významný bit), po datech odchází volitelný ParityBit a zpráva je ukončena posledním jedním, nebo dvěma StopBity.



Obr. 15. Formát zasílání zpráv na RS232

Komunikace běžně probíhá na rychlostech 9600, 19200, 38400 bajtů za sekundu.

Vzhledem k tomu, že standard RS232 využívá logických úrovní, kde logická nula je tvořena signálem s úrovní +5V až +15V a logická jednička signálem s úrovní -5 až -15V a UART (universal asynchronous receiver transmitter), v použitém procesoru logických úrovní TTL logiky (0V pro logickou nulu a +5V pro logickou 1), je nutné tyto úrovně pro možnost komunikovat po standardní RS232 převádět. K tomu slouží použitý převodník úrovní MAX232. Zapojení převodníku je blíže popsáno v kapitole 3.5.1.1.

3.4.1.1 Seriová linka v procesoru ATmega328PB

Mikrokontrolér ATmega328PB disponuje dvěma rozhraními UART (UART0 a UART1).

V zařízení je využito rozhraní UART1 pro možnost komunikace na RS232. UART je pak nastaven na rychlost 38400 bitů/s, velikost datového rámce na 8 bitů, bez Parity Bitu a s jedním Stop Bitem. Celková velikost zprávy je tedy 10 bitů, z čehož plyne, že je teoreticky možné posílat až 3840 zpráv za sekundu.

Inicializace UART rozhraní v procesoru probíhá pomocí nastavení registrů UCSR1B, UCSR1C a UBRR1.

Registr UBRR1 je 12-bitový registr, tudíž rozdělen na 2 byty. Do registru se ukládá hodnota, kterou bude dělen kmitočet z externího oscilátoru tak, aby odesílání zpráv dosahovalo požadované rychlosti. Výpočet dle následující rovnice

$$UBRR1 = \frac{f_{oscilátoru}}{16 * požadovaná\ rychlost\ přenosu} - 1$$

$$UBRR1 = \frac{16\,000\,000}{16 * 38400} - 1 = 25.041$$

Do registru se pak zapisuje pouze celočíselná hodnota, tudíž 25.

Tab. 4. Nastavení a význam bitů registru UCSR1B

Bit	Hodnota	Název	Význam
0	0	TXB8	9. bit zprávy k odeslání v při posílání 9-bitových dat
1	0	RXB8	9. bit zprávy k příjmu v při příjmu 9-bitových dat
2	0	UCSZ2	Část nastavení velikosti datového rámce (více v UCSR1C)
3	1	TXEN	Povolení vysílače
4	1	RXEN	Povolení přijímače
5	0	UDRIE	Povolení přerušení při vyprázdění datového registru
6	0	TXCIE	Povolení přerušení po odeslání zprávy (1 = ON)
7	1	RXCIE	Povolení přerušení při příjmu zprávy (1 = ON)

Tab. 5 Nastavení a význam bitů registru UCSR1C

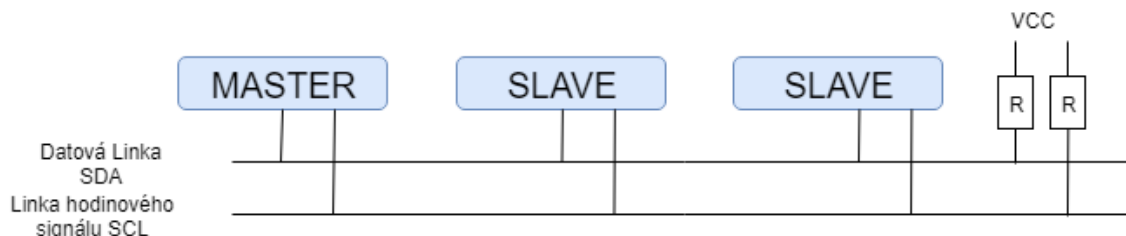
Bit	Hodnota	Název	Význam
0	0	UCPOL	Polarita hodin (0 – vysílání dat při nástupné hraně hodinového signálu, příjem dat při sestupné hraně hodinového signálu)
1	1	UCSZ0	Část nastavení velikosti datového rámce společně s bitem UCSZ2 (011 – datový rámec velikosti 8 bitů)
2	1	UCSZ1	
3	0	USBS	Nastavení StopBitů (0 – jeden StopBit)
4	0	UMP	Nastavení parity (00 – bez parity)
5	0		
6	0	UMSEL	Nastavení pracovního modu UARTu (00 – asynchronní režim)
7	0		

3.4.2 Komunikační sběrnice I2C

I2C je dvou vodičová multimaster sběrnice vyvinutá firmou Philips v devadesátých letech minulého století.

Fyzická vrstva sběrnice je tvořena dvěma vodiči - SDA a SCL. Vodič SDA je užíván pro přenos dat, vodič SCL pro časování sběrnice. Oba vodiče jsou připojené přes pull-up rezistor k napájecímu napětí, což zajistí přítomnost logické hodnoty „1“ na

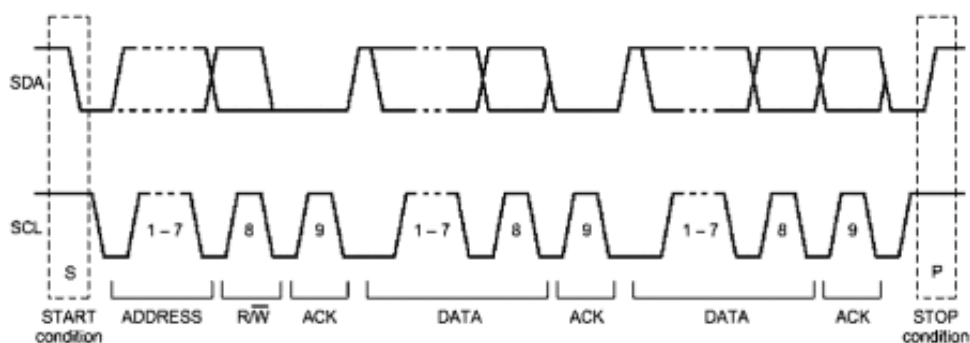
sběrnici v případě, že žádné zařízení nevysílá. Všechna zařízení na sběrnici jsou adresovaná unikátním 7, nebo 10 bitovým identifikátorem z čehož plyne i maximální počet zařízení na sběrnici (pro 7 bitovou adresu až 127 zařízení, pro 10 bitovou adresu až 1023 zařízení). Na sběrnici může vysílat vždy pouze jedno zařízení. Přenos dat probíhá v 8bitových rámcích. V každém jednom cyklu hodin SCL proběhne přenos jednoho bitu.



Obr. 16. Principiální zapojení zařízení ke sběrnici I2C

Přenos dat probíhá tak, že zařízení, které vysílá (Master) nejdříve na sběrnici vyšle StartBit (na vodiči SDA dojde ke změně signálu na logickou nulu, na vodiči SCL je logická jednička). Za StartBitem následuje adresa zařízení, se kterým chceme komunikovat v podobě 7bitů žádané adresy a následujícího bitu R/W, kterým určujeme, zda do zařízení budeme zapisovat, nebo z něho číst. Pokud zařízení, se kterým komunikujeme, zprávu přijme, vyšle na sběrnici v dalším časovém cyklu ACKBit (potvrzení o příjmu zprávy). ACKBit je definován tak, že master po vyslání 8 bitové zprávy uvolní vodič SDA, na kterém je přítomná logická jednička a slave následně úroveň změní na logickou nulu.

V situaci, kdy byla od slave zařízení požadována data, a slave odpověděl na předchozí zprávu ACKBitem, začne slave na datovou linku vysílat požadovaná data. Při správném příjmu dat master na každý rámeček odpovídá ACKBitem. Po dokončení příjmu nebo vysílání - vysílá master StopBit(na vodiči SDA dojde ke změně signálu na logickou jedničku, na vodiči SCL je logická jednička). Takto může být přenesen takřka neomezený počet 8-bitových datových rámců.



Obr. 17. Diagram přenosu dat na sběrnici I2C

3.4.2.1 I2C sběrnice v mikropočítači ATmega328PB

Sběrnice I2C je v použitém mikropočítači pojmenována zkratkou TWI (Two Wire Interface, neboli dvou vodičové rozhraní). Procesor disponuje dvěma TWI rozhraními (TWI0, TWI1), v projektu je použito rozhraní TWI0.

Inicializaci sběrnice je nutné provést nastavením registrů TWBR0, což je osmibitový registr který určuje frekvenci hodinového signálu (registr plníme hodnotou spočítanou v rovnicích uvedených níže) a TWSR0, což je registr, jehož spodní dva bity jsou využity pro předděličku frekvence.

Pro výpočet je nutné určit hodnotu frekvence hodinového signálu SCL, pro potřeby výpočtu 100kHz. Výpočet hodnot pro registry je pak následující.

$$frekvence\ SCL = \frac{frekvence\ ext.\ oscilátoru}{16 + 2(TWBR0) * (TWSR0)}$$

Pokud je zvolena hodnota obou dvou bitů TWSR pro předděličku signálu nula, Předdělička dělí signál hodnotou 1 a výpočet hodnoty pro registr TWBR0 vypadá následovně.

$$TWBR0 = \left[\frac{frekvence\ ext.\ oscilátoru}{frekvence\ SCL} - 16 \right] / 2$$

Z čehož plyne

$$TWBR0 = \frac{\left[\frac{16\ 000\ 000}{100\ 000} - 16 \right]}{2} = 72$$

Registr TWSR0 , respektive jeho 3. až 7. bit, je zároveň stavovým registrem sběrnice.

3.4.2.2 Adresování zařízení na I2C sběrnici

K jednotlivým snímačům na sběrnici je přistupováno za pomoci jejich adres, většina snímačů umožňuje adresu nastavit pomocí zapojení adresních pinů. Pro příklad snímač MCP9843 disponuje třemi adresními piny, které určují hodnotu 3 bitů adresy s nejnižší vahou. Pokud je konkrétní adresní pin připojený k logické napájecí nule, hodnota odpovídajícího bitu je „0“, naopak při připojení adresního pinu k logické napájecí „1“ je hodnoty odpovídajícího bitu „1“

Tab. 6. Adresní byte snímače MCP9843

Číslo bitu	6	5	4	3	2	1	0
Hodnota bitu	0	0	1	1	Určeno pinem A2	Určeno pinem A1	Určeno Pinem A0

Z výše uvedeného textu ,Tab. 6 a Tab. 7 je patrné, že první ze dvou snímačů má všechny adresní piny připojeny k logické napájecí nule (a jeho adresní byte je tedy roven hodnotě 0011000 binárně, tedy 18 hexadecimálně) a druhý snímač má pin „A0“ připojený k logické napájecí „1“ (a jeho adresní byte je roven 0011001 binárně, tedy 19 hexadecimálně).

V souvislosti s adresováním na sběrnici I2C pomocí 7-bitových adres je nevhodné mluvit o adrese jako o celém bytu, ale například datový list snímače MCP9843 toto připouští. Celý byte „vznikne“ až při posílání adresní části po sběrnici a to takovým způsobem, že 7 adresních bitů se posune v bytu o jeden bit doleva a na pozici „prázdného“ bitu s nejnižší hodnotou se zapíše hodnota bitu R/W (R-požadovaná operace čtení / W-požadovaná operace zápis). Následně se celý byte odešle na sběrnici.

Tab. 7. Adresy jednotlivých snímačů na sběrnici I2C

Snímaná veličina [jednotka]	Typ snímače	Adresa snímače	Poznámka
Teplota [°C]	MCP9843	0x18	Část adresy pevně daná, spodní 3 bity nastavitelné zapojením
Teplota [°C]	MCP9843	0x19	Část adresy pevně daná, spodní 3 bity nastavitelné zapojením
Intenzita osvětlení [lux]	BH1750	0x23	2 možné adresy nastavitelné zapojením
Intenzita svitu složek světla [-]	TCS34725	0x29	Adresa pevně určená výrobcem ⁽¹⁾

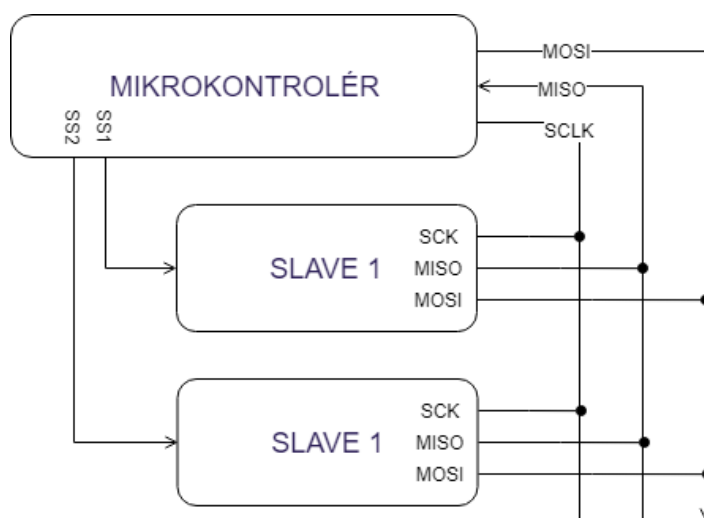
(1) – výrobce nabízí snímač buď s adresou 0x29, nebo 0x39

3.4.3 Komunikační sběrnice SPI

Sériové komunikační rozhraní SPI vyvinula společnost Motorola. Rozhraní umožňuje obousměrnou synchronní sériovou komunikaci mezi zařízeními. Využívá se hlavně pro komunikaci mezi mikrokontrolérem a připojenými komponenty – v režimu master-slave, kdy komunikaci na sběrnici řídí mikrokontrolér.

Rozhraní využívá čtyřvodičové zapojení:

1. Vodič „**SCLK**“ – využíváný pro časování rozhraní, společný pro všechna zařízení na sběrnici
2. Vodič „**SS**“ – vodič pro výběr komunikovaného zařízení, jeden vodič pro každé slave zařízení na sběrnici
3. Vodič „**MOSI**“ – (Master Out Slave In) – využíváný pro přenos dat ze zařízení konfigurovaného jako master (většinou mikrokontrolér), společný pro všechna zařízení na sběrnici
4. Vodič „**MISO**“ – (Master In Slave Out) – využíváný pro přenos dat ze zařízení konfigurovaného jako slave (převodníky analog-digitál, připojitelné periferie, externí paměti a další), společný pro všechna zařízení na sběrnici.



Obr. 18. Běžně používané zapojení sběrnice SPI

3.4.3.1 Popis komunikace SPI

Komunikace probíhá vždy pouze mezi dvěma zařízeními na sběrnici. Typicky mezi masterem, který komunikaci řídí, a slavem. Obě zařízení jsou vybaveny posuvnými registry, do kterých jsou vnitřní logikou předávána data pro odeslání.

Komunikace začíná tak, že master přitáhne signál SS zařízení se kterým chce komunikovat k úrovni logické nuly, čímž zařízení aktivuje. V tu chvíli master začne generovat hodinový signál a s hodinovým signálem začne na linku MOSI vyprazdňovat obsah posuvného registru. Zároveň může na lince MISO zároveň probíhat vyprazdňování posuvného registru slave zařízení.

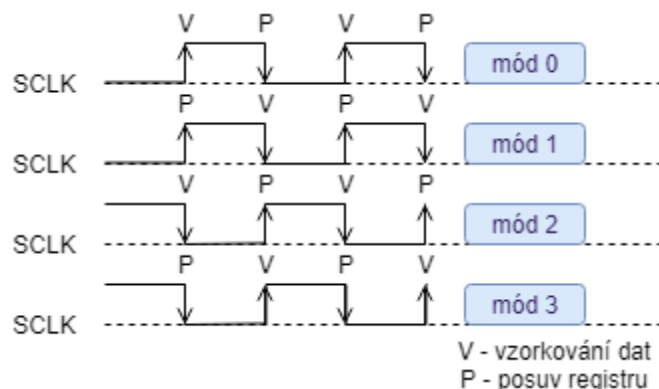
Neexistuje žádný obecný protokol, jaký datový formát (délku dat, velikost rámců, atd.) má být použit. Konkrétní implementace vždy záleží na použitých zařízeních. Výhodou tohoto způsobu komunikace je nepotřebnost adresace zařízení v datové komunikaci, což je ovšem vykoupeno nutností použití více vodičů volby zařízení SS.

Jako příklad komunikace může sloužit popis komunikace na sběrnici SPI mezi mikrokontrolérem ATMEGA 328PB a ethernet kontrolérem W5500 v tomto projektu, popsany v kapitole 3.6.3 - Práce s SPI sběrnici.

3.4.3.2 Módy časování komunikace

Protokol definuje časování komunikace, tzv. módy, a to pomocí volby polarity a fáze generovaného synchronizačního signálu „SCLK“. Kombinací voleb polarity a fáze SCLK jsou pak definovány 4 módy časování komunikace.

- Múd „0“
 - Polarita = 0, Fáze = 0
 - Data k odeslání musí být připravena v posuvném registru ještě před první periodou synchronizačního signálu. Vzorkování dat probíhá s náběžnou hranou signálu SCLK, posuv dat v posuvném registru pak se sestupnou hranou.
- Múd „1“
 - Polarita = 0, Fáze = 1
 - První náběžná hrana synchronizačního signálu je použita pro přípravu dat do posuvného registru. Nejdříve tedy probíhá posuv dat v posuvném registru při náběžné hraně SCLK. Při sestupné hraně se data vzorkují.
- Múd „2“
 - Polarita = 1, Fáze = 0
 - Data k odeslání musí být v registru opět připravena ještě před první periodou SCLK. Se sestupnou hranou se data vzorkují, s nástupnou hranou dochází k posuvu dat v registru.
- Múd „3“
 - Polarita = 1, Fáze = 1
 - Při první sestupné hraně SCLK se data posouvají v registru, při následné náběžné hraně se data vzorkují.



Obr. 19. Časování komunikace SPI v jednotlivých módech

3.4.3.3 SPI v mikrokontroléru ATMEGA 328PB

Mikrokontrolér ATMEGA 328PB disponuje dvěma master-slave rozhraními SPI (SPI0 a SPI1). Rozhraní je plně duplexní (současná komunikace oběma směry) a je možné volit mezi LSB (první odesílaný bit s nejnižší hodnotou) a MSB (první odesílaný bit s nejvyšší hodnotou) způsobem odesílání/přijímání dat. Frekvence synchronizačního signálu je odvozená od použitého oscilátoru (v zařízení použitý 16MHz) a je možné ji dělit předděličkou hodnotami 2/4/8/16/32/64 a 128. Časování SPI v mikrokontroléru může pracovat ve všech definovaných módech, tedy 0,1,2 a 3.

V zařízení je pro komunikaci s ethernet kontrolérem W5500 použité rozhraní SPI1, které se nastavuje pomocí kontrolního registru *SPCR1*. Z níže uvedené tabulky Tab. 8 je patrné, že SPI je v zařízení nastaveno jako Master, vysílání dat způsobem MSB, v časovém módu 0.

Tab. 8. Nastavení rozhraní SPI1

Bit	Hodnota	Název	Význam
0	0	SPR10	Nastavení děličky frekvence synchronizačního signálu
1	0	SPR11	Nastavení děličky frekvence synchronizačního signálu
2	0	CPHA1	Nastavení módu časování – fáze, 1-ON
3	0	CPOL1	Nastavení módu časování – polarita 1-ON
4	1	MSTR1	Režim Master = 1, Slave = 0
5	0	DORD1	Režim LSB = 1, režim MSB = 0
6	1	SPE1	Povolení rozhraní SPI1, 1 = ON
7	0	SPIE1	Povolení přerušení od SPI1, 1=ON

Do nastavení dělení frekvence kromě parametrů SPR10 a SPR11 z registru SPCR1 vstupuje ještě parametr SPIX1 z registru SPSR0. Všechny tyto parametry jsou nastaveny na nulovou hodnotu. Tím je dosaženo nastavení děličky frekvence na „/4“. Z výše uvedeného plyne, že SPI v projektu je nastaveno na frekvenci f_{spi}

$$f_{spi} = \frac{f_{osc}}{4} = \frac{16\,000\,000}{4} = 4\,000\,000\,Hz$$

$$f_{spi} = 4\,MHz$$

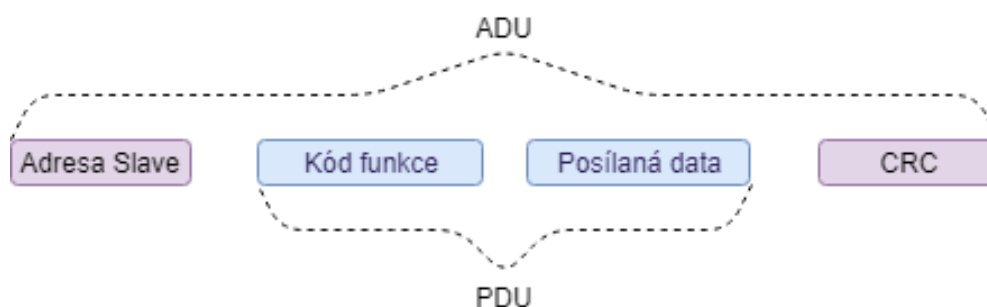
3.4.4 Modbus

Komunikační protokol MODBUS vyvinula firma Modicon v roce 1979. Jedná se o protokol typu klient-server, kterým je umožněno jednoduše a organizovaně přenášet data skrze sériová rozhraní. Implementace MODBUS je realizovaná na aplikační vrstvě modelu ISO/OSI. Jako fyzické vrstvy modelu jsou pro implementaci používány například: Sériová rozhraní RS232, RS485 a Ethernet.

MODBUS je protokol typu „požadavek-odpověď“ a požadované úkony jsou prováděny pomocí definovaných funkcí, respektive funkčních kódů.

Protokol je volně dostupný pro různé výrobce elektronických zařízení a tak vzniká množství variant upravených pro konkrétní potřeby daných firem.

Celá posílaná zpráva se skládá z více celků. Základní popis zprávy je na Obr. 20.



Obr. 20. Formát zprávy protokolu MODBUS

Z obrázku je patrné, že zpráva se skládá z PDU (protocol data unit), který je nezávislý na přenosovém médiu – je tedy jedno, zda-li data přijdou na MODBUS RTU, nebo MODBUS TCP/IP. PDU je vždy stejné. PDU je pak dle použitého přenosového média doplněn o další informace. Sloučením PDU s aditivními informacemi se skládá ADU (aplication data unit).

PDU lze rozdělit podle typu odesílané zprávy na:

- Požadavek – zpráva žádající dotazované zařízení
- Odpověď – zpráva s požadovanými informacemi pro tazatele
- Chyba – zpráva pro tazatele o nemožnosti vykonat požadovanou operaci

Velikost PDU je v protokolu omezena na 253 Bytů.

3.4.4.1 Funkce implementované v zařízení a jejich forma

Jak již bylo řečeno, PDU je nezávislé na přenosovém médiu, tudíž jeho tvar i zpracování v mikrokontroléru může být vždy stejné. PDU se ale mění podle použité funkce.

Tab. 9. Funkce MODBUS

Kód funkce	Název	Implementace v zařízení	Popis
0x01	Read Coil Status (čti stav cívky)	Ne	
0x02	Read Input Status (čti stav vstupu)	Ne	
0x03	Read Holding Registers (čti registry)	Ano	Požadavek na čtení oblasti z paměťového prostoru
0x04	Read Input Registers (čti vstupní registry)	Ne	
0x05	Force Single Coil (nastav stav jedné cívky)	Ne	
0x06	Preset Single Register (nastav jeden registr)	Ano	Požadavek na nastavení jednoho registru v paměťovém prostoru
0x10	Force Multiple Coils (nastav stav více cívek)	Ano	Požadavek na nastavení více registrů v paměťovém prostoru
0x11	Preset Multiple Registers (nastav stav více registrů)	Ne	

V následujících kapitolách budou popsány jednotlivé funkce implementované v zařízení. V každé funkci je komunikováno na zařízení s adresou, neboli Slave ID, 0x01. Všechny hodnoty jsou pak uváděny v hexadecimálním tvaru tak, jako v předchozí tabulce.

3.4.4.2 Popis a příklad funkce 0x03

Požadavek klienta na server (zařízení). Jsou vyžadovány 2 registry od adresy 0x10 v paměťovém prostoru. Při úspěšném zápisu server odpovídá zprávou uvedenou v tabulce.

Tab. 10. Formát zprávy Modbus, funkce 0x03

Dotaz		Odpověď	
Slave ID	0x01	Slave ID	0x01
Kód funkce	0x03	Kód funkce	0x03
Adresa prvního registru	0x00	Počet následujících data bytů	0x04
	0x10	Obsah 1. registru	0x1A
Počet žádaných registrů	0x00		0x1A
	0x02	Obsah 2. registru	0x1B
CRC kód	0xC5		0x1B
	0xCE	CRC kód	0x96
			0x17

3.4.4.3 Popis a příklad funkce 0x06

Požadavek klienta na server (zařízení). Zapisujeme do registru na adrese 0x0010, hodnotu 0x0002. Při úspěšném zápisu server odpovídá zcela stejnou zprávou.

Tab. 11. Formát zprávy Modbus, funkce 0x06

Dotaz		Odpověď	
Slave ID	0x01	Slave ID	0x01
Kód funkce	0x06	Kód funkce	0x06
Adresa registru	0x00	Adresa registru	0x00
	0x10		0x10
Zapisovaná hodnota	0x00	Zapisovaná hodnota	0x00
	0x02		0x02
CRC kód	0x09	CRC kód	0x09
	0xCE		0xCE

3.4.4.4 Popis a příklad funkce 0x10

Požadavek klienta na server (zařízení). Zapisujeme do 3 registrů od adresy 0x0010, hodnoty z tabulky. Při úspěšném zápisu server odpovídá zprávu uvedenou v tabulce.

Tab. 12. Formát zprávy Modbus, funkce 0x10

Dotaz		Odpověď	
Slave ID	0x01	Slave ID	0x01
Kód funkce	0x10	Kód funkce	0x10
Adresa prvního registru	0x00	Adresa prvního registru	0x00
	0x10		0x10
Počet zapisovaných registrů	0x00	Počet zapisovaných registrů	0x00
	0x03		0x03
Počet následujících data bytů	0x00	CRC kód	0x81
	0x06		0xCD
Hodnota registru 1	0x1A		
	0x1A		
Hodnota registru 2	0x1B		
	0x1B		
Hodnota registru 3	0x1C		
	0x1C		
CRC kód	0x97		
	0x72		

3.4.4.5 Výpočet CRC

Pomocí kontrolního součtu CRC (cyclic redundancy check) je kontrolována správnost a úplnost doručené zprávy. Princip kontroly spočívá ve vypočtení kontrolního součtu pro odesílanou zprávu v odesílateli, následného odeslání zprávy včetně CRC příjemci. V příjemci se z přijatých dat CRC znovu nezávisle vypočte a porovná s přijatou hodnotou CRC. Pokud jsou takto vypočtené CRC shodné, zpráva přišla korektně a úplně.

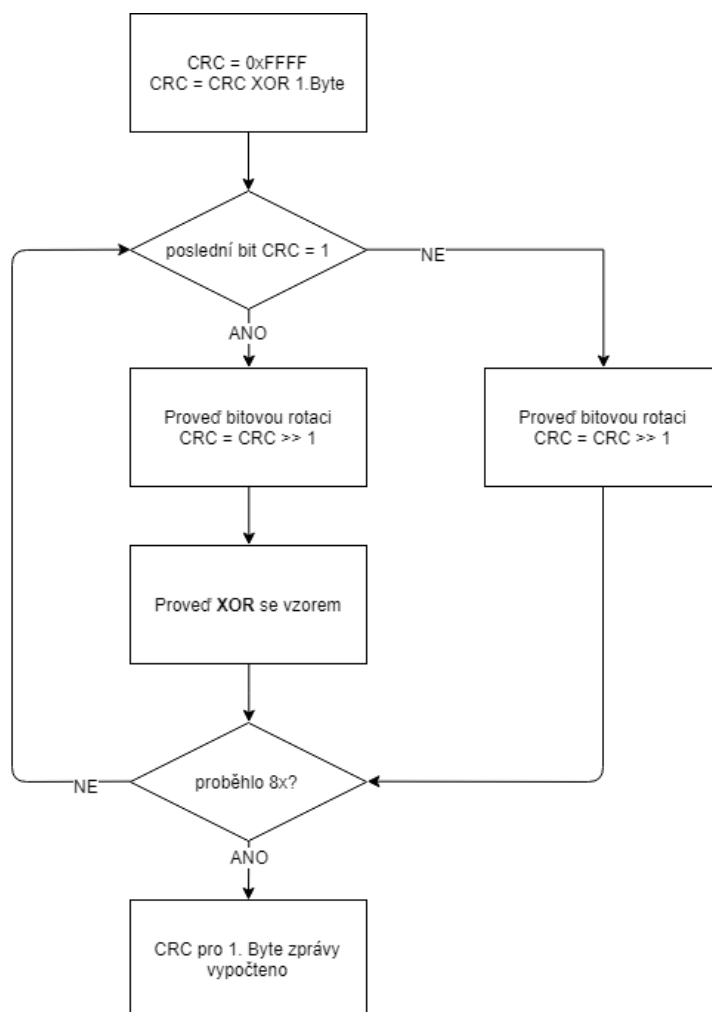
V protokolu Modbus je použito 16 bitového CRC. Výpočet protokolu je možný pomocí pomocí tabulky (tento způsob je méně náročný pro procesor, díky uložené tabulce však zabere více paměti), nebo smyčky v programu (tento způsob sice zabere více procesorového času, ale je názornější pro pochopení výpočtu).

Pro výpočet se využívá logické funkce XOR, neboli exklusivní disjunkce. Z níže uvedené tabulky je patrné, že funkce XOR je pravdivá právě když je jeden z jejích vstupů rozdílný od ostatních.

Tab. 13. Pravdivostní tabulka XOR

Vstup A	Vstup B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Výpočet se provádí pro celou zprávu následujícím způsobem: Na začátku výpočtu má CRC hodnotu 0xFFFF. Pro první bajt zprávy je proveden XOR s CRC. Následuje 8 krát opakovaná sekvence,



Obr. 21. Sekvence pro výpočet CRC pro 1 byte zprávy

Takto výpočet pokračuje pro všechny byty zprávy. Pro další byty již probíhá XOR aktuálního CRC s daným bytem.

3.4.4.6 Modbus RTU

Modbus RTU je v zařízení implementován pro komunikaci na seriové lince. Tvar ADU odpovídá vyobrazení na Obr. 20. Začátek zprávy je detekován prvním znakem. Mezi znaky nesmí být mezera delší než 3,5 znaku. Pokud je mezera delší než 3,5 znaku, znamená to, že nastal konec zprávy.

3.4.4.7 Modbus TCP/IP

Modbus TCP/IP je v zařízení implementován pro komunikaci po ethernetu. Tvar ADU je pozměněn. ADU již neobsahuje kontrolní součet, přibyla pak hlavička MBAP (ModBus Application Protocol). Jednotlivé části MBAP

- Číslo transakce – stejné v dotazu i odpovědi, inkrementuje se při každé další zprávě
- ID protokolu – konstantní hodnota 0x0000
- Délka zprávy – počet následujících bytů zprávy
- Adresa Slave – adresa zařízení, jemuž je zpráva určena



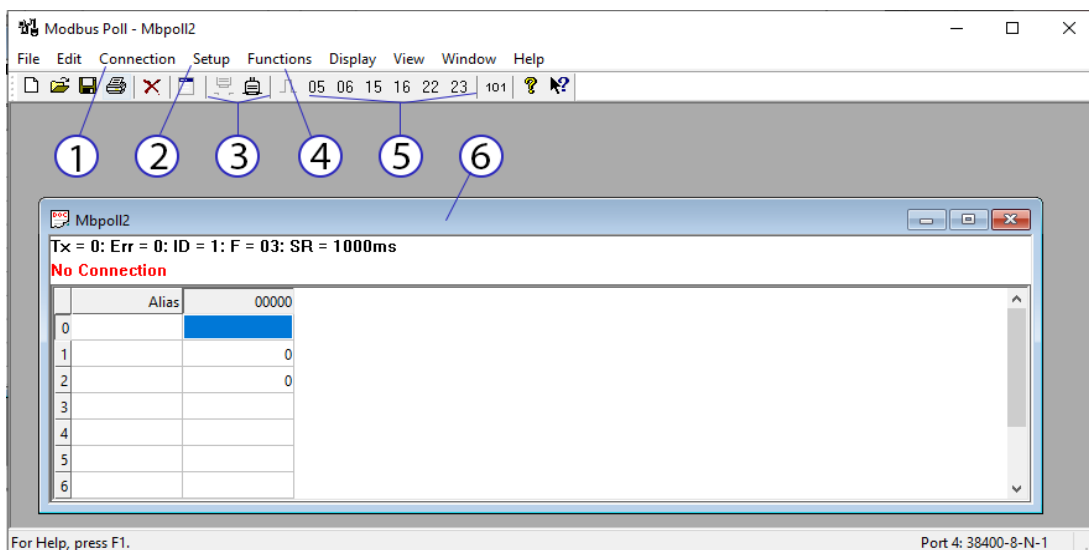
Obr. 22. Tvar zprávy Modbus TCP

3.4.5 Komunikace se zařízením pomocí Modbus klienta

Pro ověření komunikace se zařízením byl zvolen program Modbus Poll volně dostupný ke stažení v omezené verzi na webové adrese https://www.modbustools.com/modbus_poll.html. Omezení verze spočívá v omezení doby připojení k zařízení na dobu 10 minut. Poté dojde k odpojení a program je nutné restartovat. Dalším omezením je 30 denní zkušební lhůta, po které nelze navázat spojení.

Uvedený SW podporuje mnoho typů Modbus komunikace. Pro použití se zařízením jsou důležité funkce Modbus TCP/IP a Modbus RTU, které jsou implementovány dle standardu.

3.4.5.1 Prostředí Modbus Poll



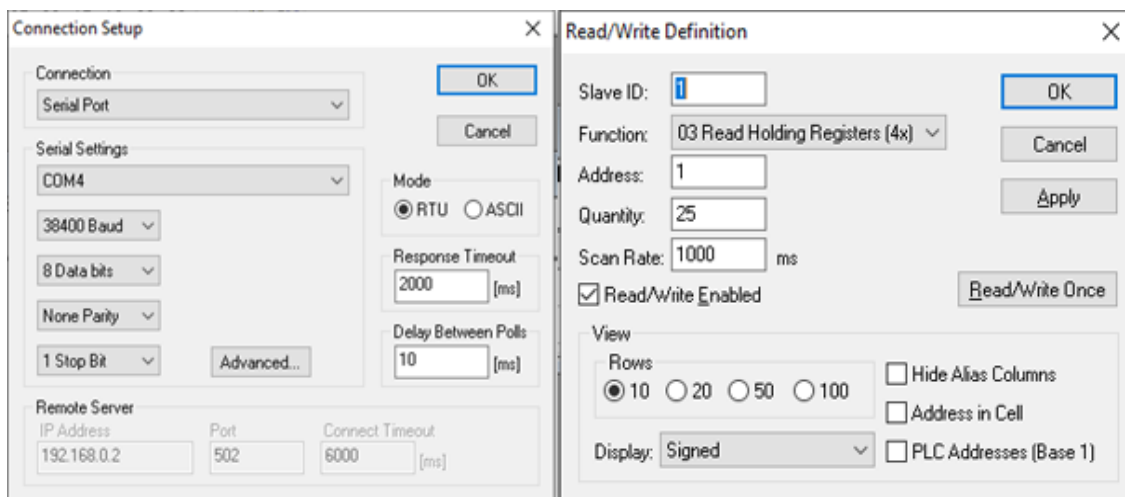
Obr. 23. Snímek prostředí programu Modbus Poll

Důležitými prvky ovládacího prostředí programu jsou:

1. Záložka „Connection“, pomocí níž se zahajuje spojení
2. Záložka „Setup“, ve které se nastavují parametry vyčítání a zápisů
3. Ikony rychlého připojení (vpravo) a rychlého odpojení (vlevo) komunikace
4. Záložka „Functions“, která zpřístupňuje všechny implementované komunikační funkce
5. Rychlý výběr jednotlivých často používaných funkcí
6. Okno aktuálního spojení

3.4.5.2 Ověření komunikace Modbus RTU

Pro komunikaci se zařízením je nutné nastavit v menu „Connection Setup“ příslušný komunikační port, ke kterému je zařízení připojeno, komunikační rychlost (38400 baudů), formát zprávy na 8 data bitů, bez parity bitu a s jedním stop bitem. V menu „Read/Write Definition“ pak volíme Slave ID (pro komunikaci se zařízením rovno 1), příslušnou funkci (0x03 – čtení registrů), adresu registrů odkud započne čtení a počet čtených registrů. Parametr Scan Rate určuje s jakou periodou bude do zařízení vysílán dotaz.



Obr. 24. Nastavení komunikace v režimu Modbus RTU

Po úspěšném spojení se zařízením započne vyčítání dat z oblasti určené nastavením. Na Obr. 25 je vidět výřez přijímaných dat. Ze stavové lišty v horní části okna jemožné vyčíst počet odeslaných dotazů (Tx=23), počet nedoručených, nebo chybných odpovědí od zařízení (Err = 0), Identifikátor zařízení (ID=1), Používanou funkci Modbus (F=03) a nastavenou periodu vyčítání (SR = 1000ms).

V datové části je na řádku 8 celočíselná hodnota teploty snímané na jednotce osvětlení a na řádku 9 je část snímané teploty za desetinnou čárkou. Toto registrové rozložení neodpovídá finální podobě mapy registrů.

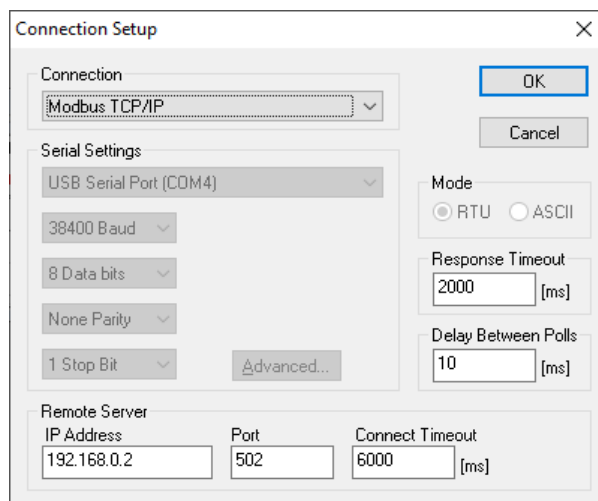
	Alias	00000	Alias	00010	Alias	00020
8		20		0		
9		50		0		

Obr. 25. Přijatá data po Modbus RTU

Všechny implementované funkce (viz Tab. 9) byly odzkoušeny při komunikaci přes Modbus RTU.

3.4.5.3 Ověření komunikace Modbus TCP/IP

Pro komunikaci přes Modbus TCP/IP je nutné nastavit v menu „Connection Setup“ jako druh připojení Modbus TCP/IP, cílovou IP adresu zařízení (parametr „IP Address“ = 192.168.0.2), příslušný komunikační port (parametr Port = 502). Nastavení menu „Read/Write Definitions“ zůstává stejné jako při předchozím testu komunikace přes Modbus RTU.



Obr. 26. Nastavení komunikace v režimu Modbus TCP/IP

Po úspěšném připojení klienta (Modbus Poll na PC) k server (řídící jednotka osvětlení) započne cyklické vyčítání dat z jednotky.

Všechny implementované funkce (viz Tab. 9) byly odzkoušeny při komunikaci přes Modbus RTU.

3.5 Konstrukce zařízení

Při návrhu konstrukce zařízení byla zohledněna zvolená modulární koncepce, požadavky na teplotní stabilitu, odolnost i dobré světelné vlastnosti. Zařízení je vyrobeno jako robustní a zároveň kompaktní.

3.5.1 Elektrická konstrukce

Po odzkoušení jednotlivých komponent na první vývojové sestavě a nastudování zapojení byly navrženy 3 desky plošných spojů. Návrh byl zhotoven ve studentské licenci programu Eagle verze 9.3.0. Při návrhu rozměrů desek byl brán ohled na zvolenou konstrukci a použití nakupovaných dílů. V následujících podkapitolách jsou popsány zapojení důležitých komponent desek, přehled navržených desek a popis jejich konektorů.

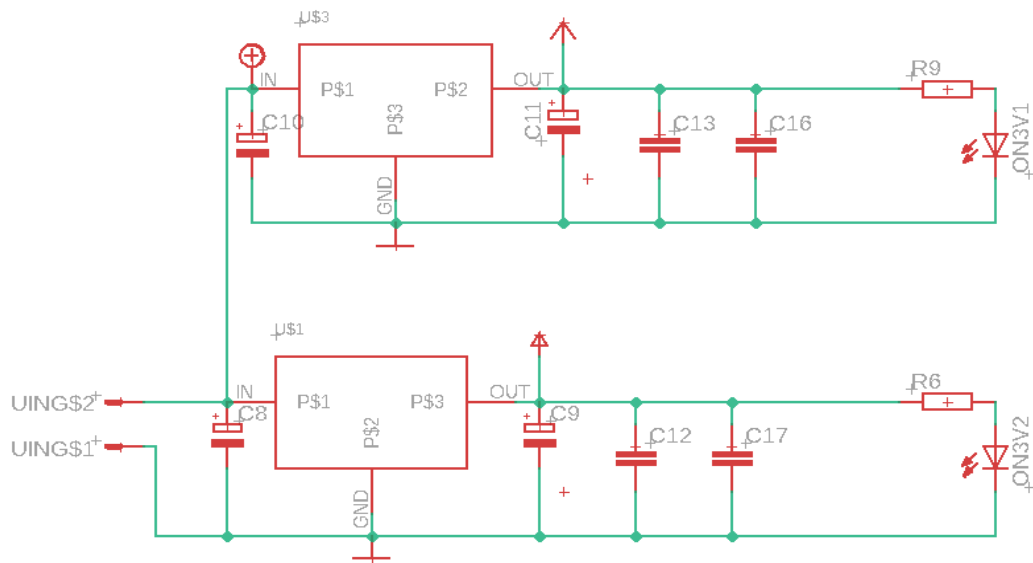
3.5.1.1 Zapojení seriové linky

Pro převod úrovní na seriové lince je využit obvod MAX232. Linky Rx a Tx z procesoru jsou připojeny k obvodu MAX232, na kterém dochází k převodu úrovní. Na tyto linky jsou připojené diody napájené z napájecí větve 3,3V. Pokud na komunikačních linkách nedochází k přenosu dat, jsou linky v logické úrovni +5V, diody přes diody tudíž neprochází žádný proud. Pokud na lince dojde k přenosu bitu, logická

3.5.1.2 Napájecí obvody

Kondenzátorová obestavba stabilizátorů vychází ze standardního zapojení uváděného výrobcem. Pro lepší filtraci výstupního napětí byly ke stabilizátorům přidány vždy 2 kondenzátory hodnoty 100nF.

49



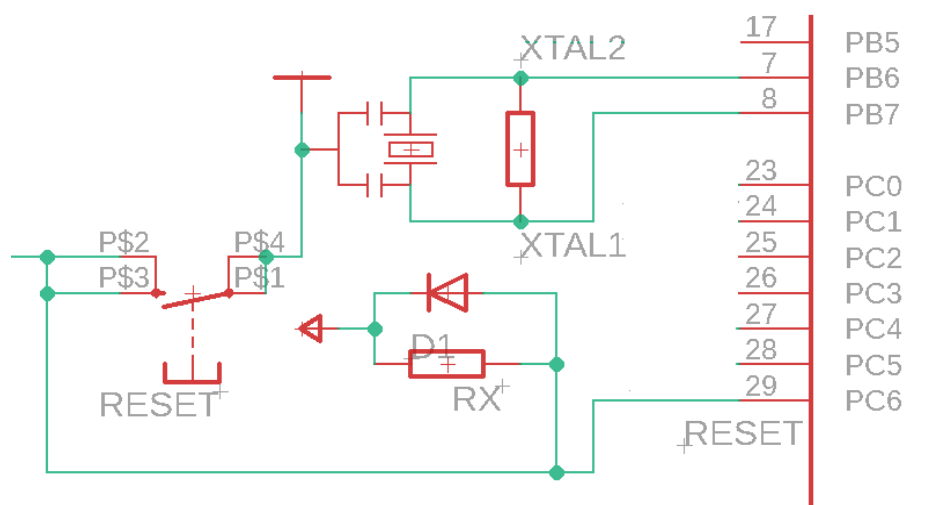
Obr. 28. Napájecí obvody zařízení

3.5.1.3 Připojení externího oscilátoru a resetovací obvod

Jako zdroj hodinového signálu pro procesor je použitý externí oscilátor s rezonanční frekvencí 16MHz. Společně s rezistorem s hodnotou $1\text{M}\Omega$ tvoří rezonanční článek.

Resetovací obvod je tvořen spínacím tlačítkem, ochrannou diodou a odporem. Při běžném provozu toto zapojení zajišťuje napětí 5V na pinu RESET procesoru. Při stisknutí tlačítka dojde k přizemnění a obvod seresetuje.

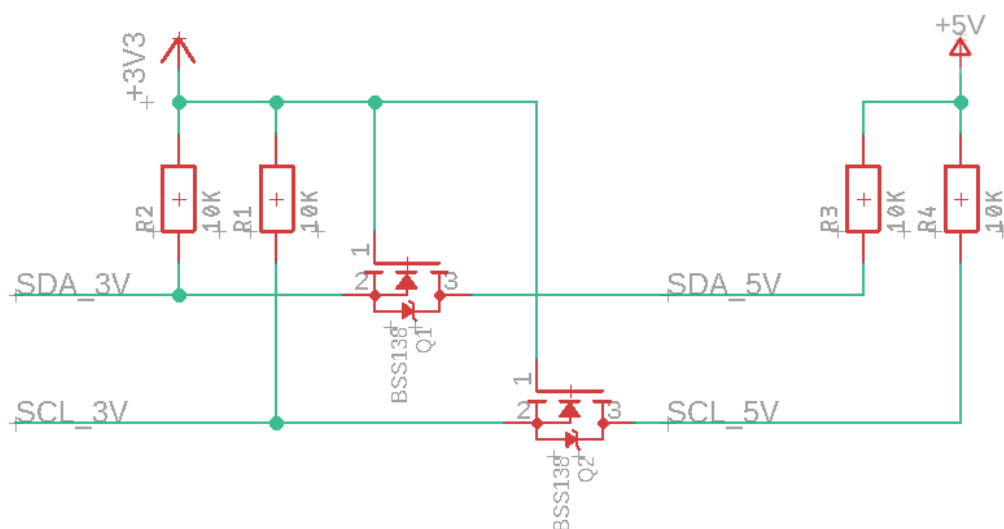
Resetovací signál je zároveň přiveden na programovací konektor a je programátorem využíván k resetování procesoru při jeho programování.



Obr. 29. Připojení resetovacího obvodu a externího oscilátoru

3.5.1.4 Zapojení I2C sběrnice

Sběrnice I2C v použitém procesoru pracuje na 5V logické úrovni. Většina snímačů připojených ke sběrnici pracuje s 3,3V úrovní. I přes to, že některé ze snímačů jsou 5V logice tolerantní, je v části světla pomocí tranzistorů BSS133 realizován převod z 5V úrovně na 3,3V úroveň a až poté jsou připojeny ke komunikačním pinům snímačů. Jednotlivé linky SDA a SCL jsou dle standardu I2C sběrnice přitaženy pomocí pull-up rezistorů k napájecímu napětí.

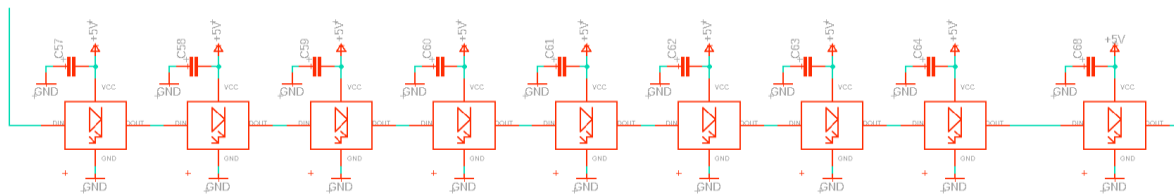


Obr. 30. Zapojení linek SDA a SCL komunikační sběrnice I2C

3.5.1.5 Zapojení inteligentních LED diod

Jak již bylo zmíněno částí zabývající se principem funkce inteligentních LED diod, lze tyto jednoduše skládat do série bez nutnosti řešit složité napájení a ovládání. Před napájecí vstup každé diody je dle doporučení výrobce umístěn keramický kondenzátor hodnoty 1uF. Do datového vstupu první diody v sérii je přiveden výstup z procesoru, z datového výstupu diody je pak datová linka přivedena na datový vstup další diody v sérii. Vnitřní logika diod se pak stará o přenos potřebných dat.

Na Obr. 31 je zobrazené zapojení posledních 8 diod v sérii 64 diod na desce plošného spoje a zcela na konci datové linky je připojena kontrolní dioda. Princip zapojení spoléhá na to, že pokud bude nefunkční jedna dioda na desce plošných spojů, potřebná data pro další diody přes ní neprojdou dál, tudíž nebude svítit ani kontrolní dioda, z čehož je následně možné vzdáleně diagnostikovat chybu.



Obr. 31. Zapojení inteligentních LED diod

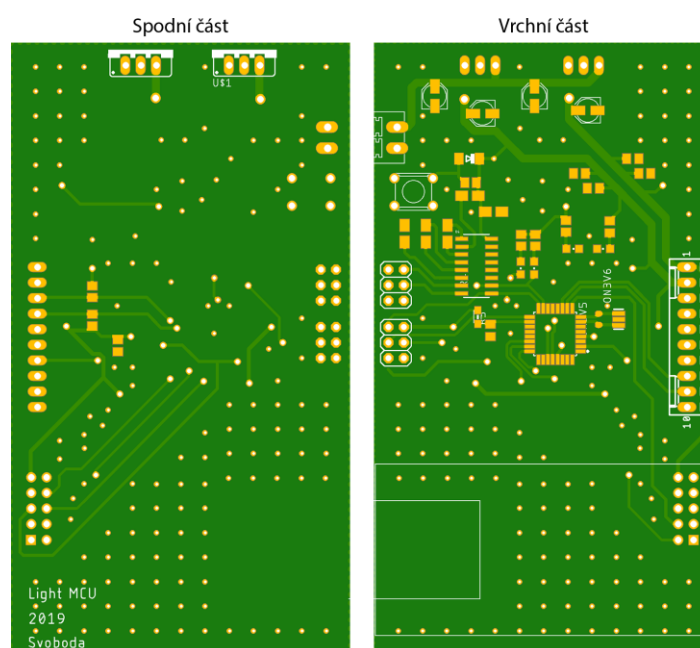
3.5.1.6 Desky plošných spojů

Desky plošných spojů vznikly ve dvou verzích. První verze byly bez umístění kontrolní diody a s chybami v komunikační části SPI rozhraní. Desky jsou 2 vrstvé, tloušťka desky je vždy 1,6mm.

Výroba plošných spojů dle dodaných Gerber dat byla zajištěna externí firmou, osazení součástek bylo provedeno samostatně – ručně. Vzhledem k použitým pouzdrům součástek lze ručním pájením spolehlivě osadit všechny součástky.

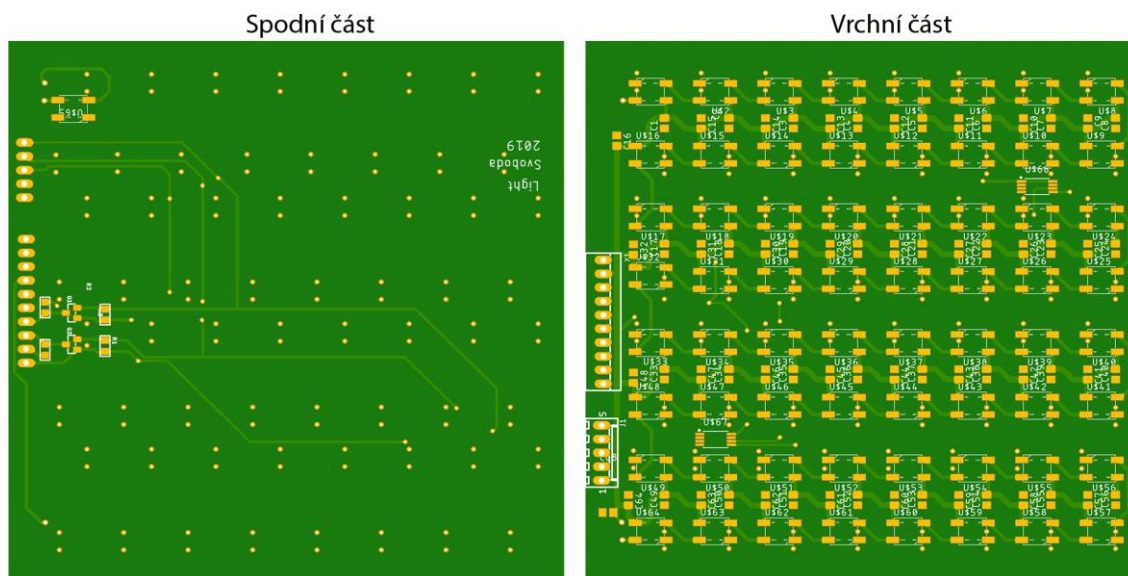
Níže uvedené obrázky desek plošných spojů nejsou v měřítku.

Deska plošných spojů řídicí části zařízení má rozměry 100x56mm Na Obr. 32 je vidět naznačený výřez pro konektor RJ45 ethernetového modulu. Na desce se nachází konektor pro připojení Ethernet modulu, 2-pinový napájecí konektor, 6-pinový programovací konektor, 6-pinový konektor pro připojení na seriovou linku a desetipinový konektor pro připojení části světla. Z důležitých elektronických součástek: procesor, převodník sériové linky, lineární stabilizátory napětí a ethernet modul.



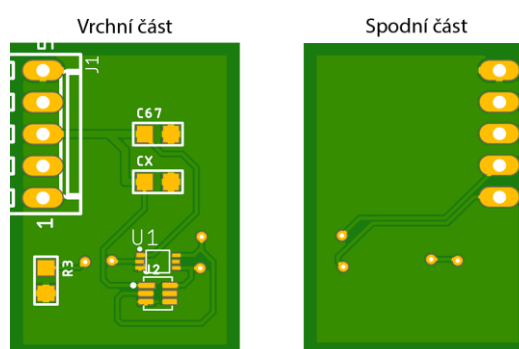
Obr. 32. Deska plošných spojů řídicí části

Deska části světla má rozměry 100x100mm. Obsahuje matici inteligentních LED diod 8x8 a jednu kontrolní diodu umístěnou na spodní straně, dvě teplotní čidla umístěná v 1. a 3. kvartálu, 10-pinový konektor pro připojení k řídicí jednotce a 5-pinový konektor pro připojení desky čidel.



Obr. 33. Deska plošných spojů části světla

Deska kontrolních snímačů má rozměry 18x25mm. Obsahuje snímač intenzity osvětlení a snímač jednotlivých složek světla. Pomocí 5-pinového konektoru je připojena k desce světla.



Obr. 34. Deska plošných spojů kontrolních snímačů

3.5.1.7 Jednotlivé konektory DPS a jejich zapojení

- DPS řídicí části
 - Napájecí konektor X1:

Tab. 14. Napájecí konektor X1

Pin	Signál	Popis
1	VCC	Napájecí napětí
2	GND	Napájecí zem

- Konektor sériové linky X2:

Tab. 15. Konektor sériové linky X2

Pin	Signál	Popis
1	GND	Napájecí zem
3	Rx	Linka příjmu dat po RS232
5	Tx	Linka vysílání dat po RS232
2, 4, 6	n.c.	Nezapojeno

- Programovací konektor X3:

Tab. 16. Programovací konektor X3

Pin	Signál	Popis
1	MISO	Linka pro přenos dat z procesoru
2	VCC	Napájecí napětí
3	SCK	Hodinový signál
4	MOSI	Linka pro přenos dat do procesoru
5	RST	Reset
6	GND	Napájecí zem

- Konektor připojení části světla X4_1:

Tab. 17. Konektor připojení části světla X4_1

Pin	Signál	Popis
1	3V3	Napájecí napětí 3,3V
2	5V	Napájecí napětí 5V
3	SCL_5V	Hodinový signál I2C sběrnice v 5V úrovních
4	SDA_5V	Datová linka I2C sběrnice v 5V úrovních
5-6	GND	Napájecí zem
7	DATAOUT	Data pro ovládání inteligentních LED
8	-	IO rezerva pin PD7 procesoru
9	-	IO rezerva pin PD6 procesoru
10	n.c.	Nezapojeno

- Konektor připojení Ethernet kontroléru X5:

Tab. 18. Konektor připojení Ethernet kontroléru X5

Pin	Signál	Popis
1	3V3	Napájecí napětí 3,3V
2	5V	Napájecí napětí 5V
3	MISO	Linka pro přenos dat do procesoru
4	GND	Napájecí zem
5	MOSI	Linka pro přenos dat do ethernetového kontroléru
6	RESET	Resetovací linka
7	SS	Slave Select
8	INT	Nezapojeno
9	SCLK	Hodinový signál
10	n.c.	Nezapojeno

- DPS části světla

- Konektor připojení k řídicí části: Zapojení konektoru X4_2 pro připojení DPS světla k DPS řízení je stejné jako u konektoru X4_1 na DPS řízení
- Konektor pro připojení části snímačů X6_1:

Tab. 19. Konektor připojení části snímačů X6_1

Pin	Signál	Popis
1	GND	Napájecí zem
2	5V	Napájecí napětí 5V
3	3V3	Napájecí napětí 3V
4	SCL_3V	Hodinový signál I2C sběrnice v 3,3V úrovních
5	SDA_3V	Datová linka I2C sběrnice v 3,3V úrovních

- DPS snímačů

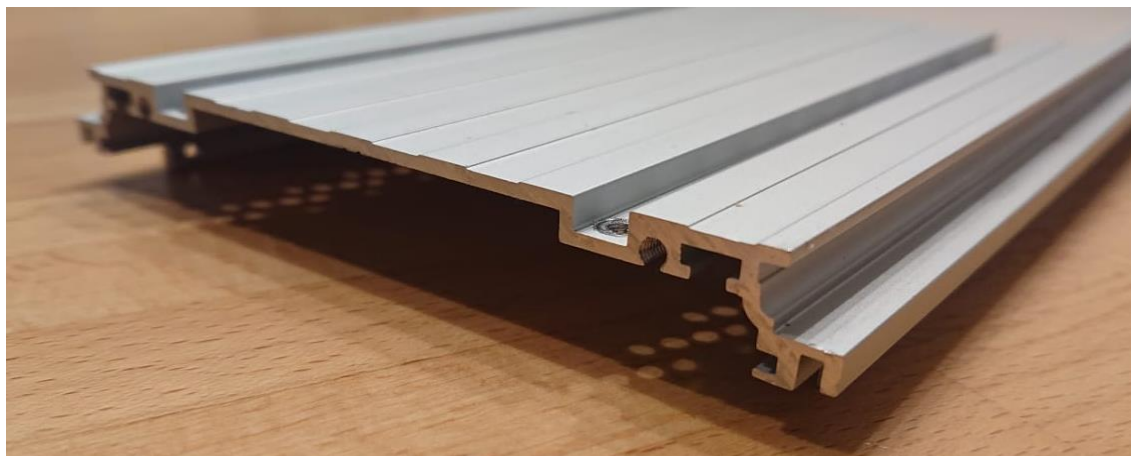
- Zapojení konektor X6_2 pro připojení DPS snímačů k DPS světla je totožné se zapojením konektoru X6_1

3.5.2 Mechanická konstrukce

Pro dobrou odolnost a dobrý rozptyl tepla z obou částí světla byl zvolen jako hlavní nosná část šasí hliníkový profil, jako protikus profilu byl použit ohýbaný hliníkový plech. Bočnice částí zařízení jsou vyrobeny z ABS plastu a průhled pro světlo ze speciálního optického plexiskla. Jednotlivé materiály a jejich zpracování je popsáno v následujících podkapitolách.

3.5.2.1 Hliníkové šasí

Hliníkový původně určený součást kazety rackového systému. Profil je odolný i díky povrchové úpravě eloxováním - eloxování je proces anodické oxidace hliníku, při kterém vzniká na povrchu upravovaného dílce vrstva oxidu hliníku ve vrstvě desítek mikrometrů, která po ukončení procesu již dále nekoroduje. Díky eloxování získá výsledný povrch jednotlý vzhled a vzhledem k vysoké tvrdosti vrstvy oxidu hliníku i velkou mechanickou odolnost.

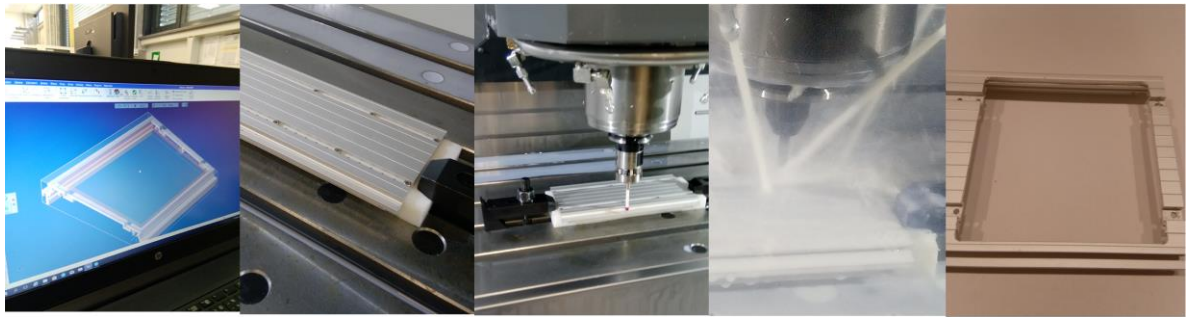


Obr. 35. Hliníkový profil jako základ pro šasí

Jako protikus pro uzavření profilu byl zvolen hliníkový plech tloušťky 2mm bez povrchové úpravy.

U světelné části bylo nutné v profilu vytvořit otvor jako průhled pro svítivé diody. Byl zvolen postup zkreslení 3D modelu a následné frézování na 3 osé fréze. Na Obr. 36 je zobrazen postup obrábění:

1. Tvorba drah pro obrábění
2. Uchycení profilu
3. Načítání nulových poloh
4. Obrábění
5. Výsledek obrábění



Obr. 36. Jednotlivé kroky obrábění hliníkového profilu

3.5.2.2 Bočnice zařízení

Bočnice zařízení jsou vyrobeny z ABS plastu tloušťky 3mm, černé barvy, který je vhodný pro obrábění pomocí laseru. Je odolný při použití do teplot až 120°C.

Z 3D návrhu obou částí byly vytvořeny 2D výkresy pro obrábění pomocí laseru. Výroba byla zadána externí firmě.

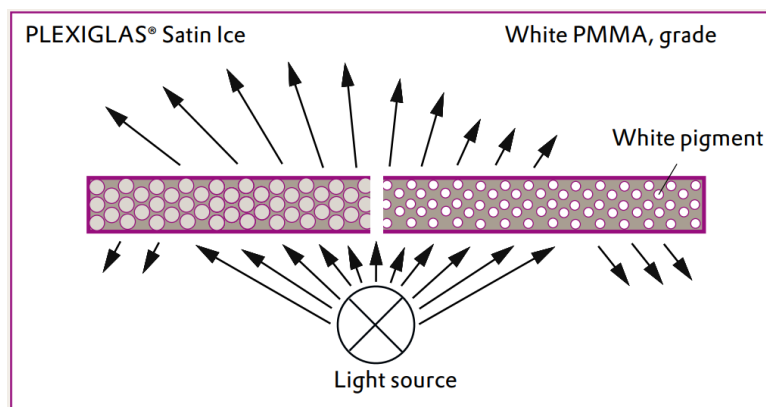
3.5.2.3 Průhled pro svítivé diody

Při volbě materiálu pro průhled svítivých diod bylo nutné dodržet následující parametry:

- Vysoká difuze a nízká odrazivost světla
- Vysoká světelná propustnost materiálu
- Tepelná odolnost
- Odolnost proti mechanickému poškození

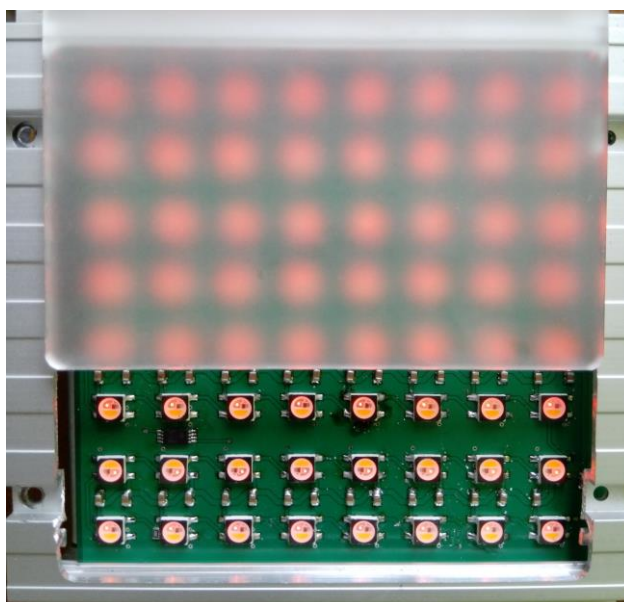
Výše uvedené parametry splňuje materiál Plexiglas® Satinice. Konkrétně typ Crystal 0F00DC tloušťky 3mm, což je oboustraně pískovaná varianta plexiskla čiré barvy s propustností světla 82%. Materiál je tepelně odolný až do 120°C.

Uvedený materiál je běžně používán v exteriérové světelné reklamě, kde se pro vysokou životnost předpokládá vysoký životnost, odolnost vůči povětrnostním vlivům i UV zářením.



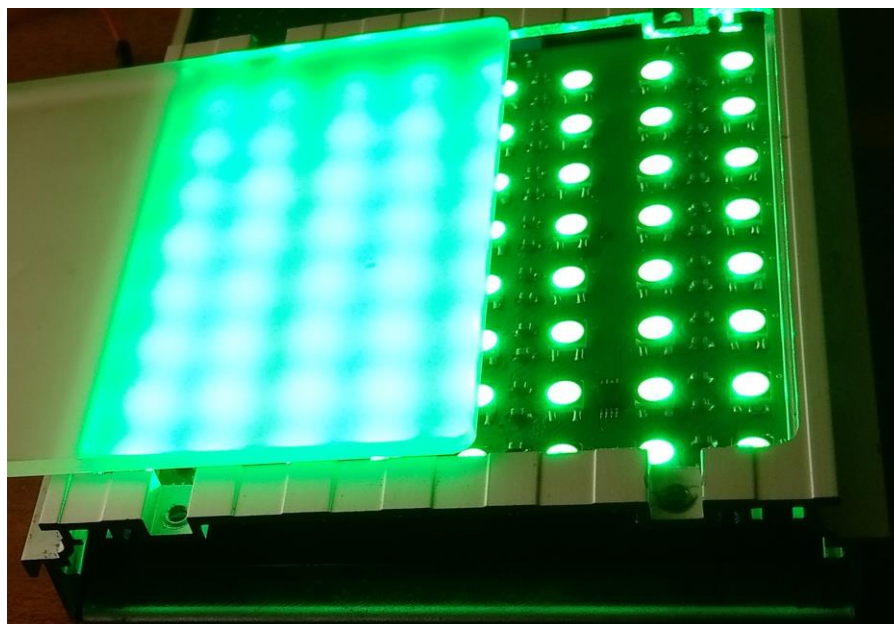
Obr. 37. Porovnání vlastností Plexiglas Satinice s běžným plexisklem [15]

Výše uvedené vlastnosti je možné demonstrovat na přiložených snímcích. Na snímcích je jasně patrná vysoká difusivita materiálu společně s vysokou propustností světla.



Obr. 38. Demonstrace vlastností materiálu – nízká intenzita, červená barva

Na Obr. 38 jsou jasně patrné definované vlastnosti. Na inteligentních LED rozsvícena pouze červená část a to na nejnižší možné intenzitě svitu (1 z 255).

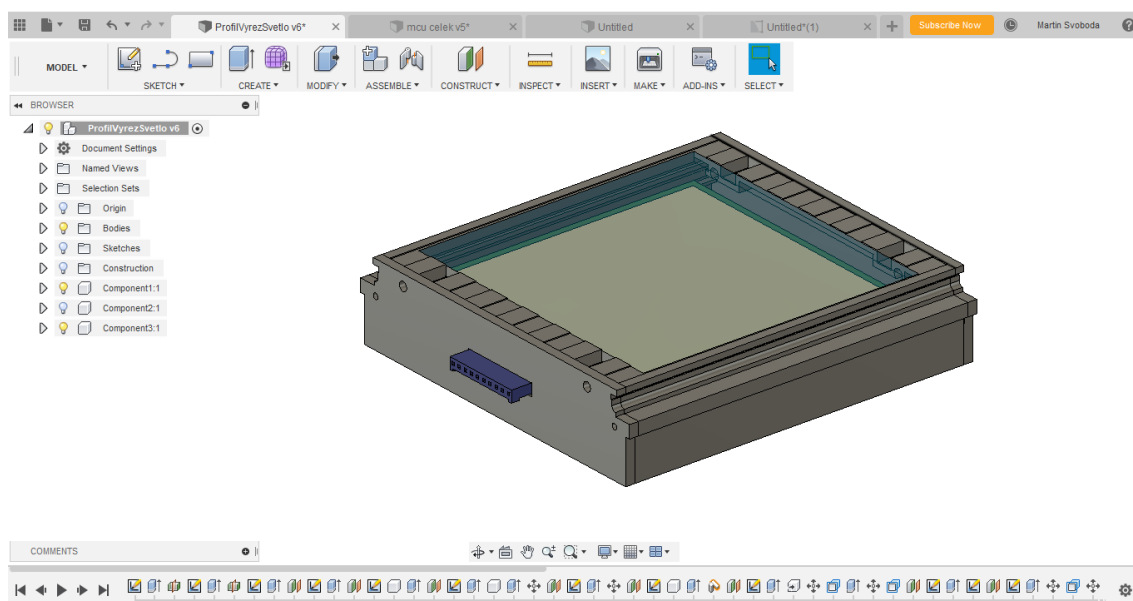


Obr. 39. Demonstrace vlastností materiálu - vysoký svit, zelená barva

Na Obr. 39 jsou pak uvedené vlastnosti demonstrovány za vysoké intenzity svitu zelené části inteligentních LED (zvolená intenzita 255 z 255).

3.5.2.4 3D návrh

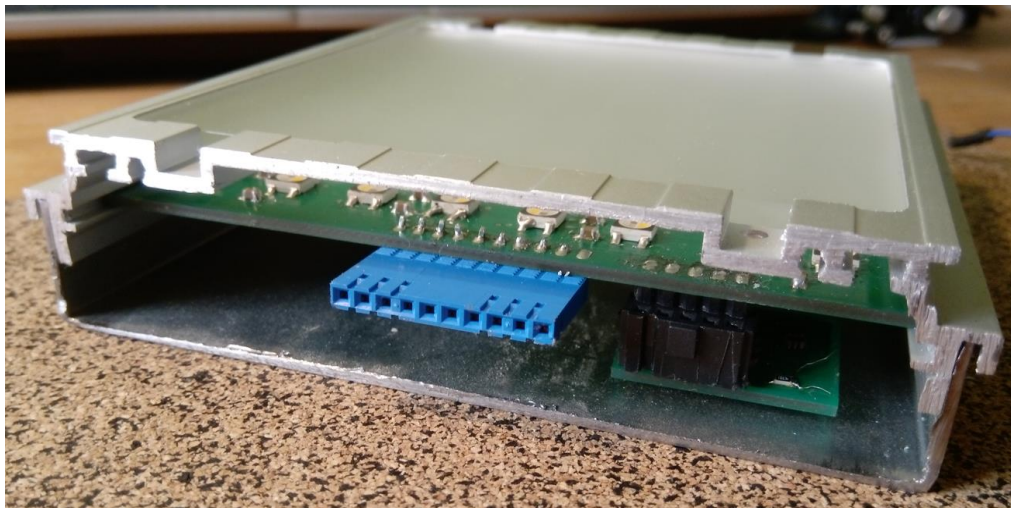
Za účelem tvorby podkladů pro opracování hliníkového profilu a výrobu ostatních částí bylo šasi pro obě části zařízení navrženo v programu Autodesk Fusion 360 volně dostupné ve studentské verzi.



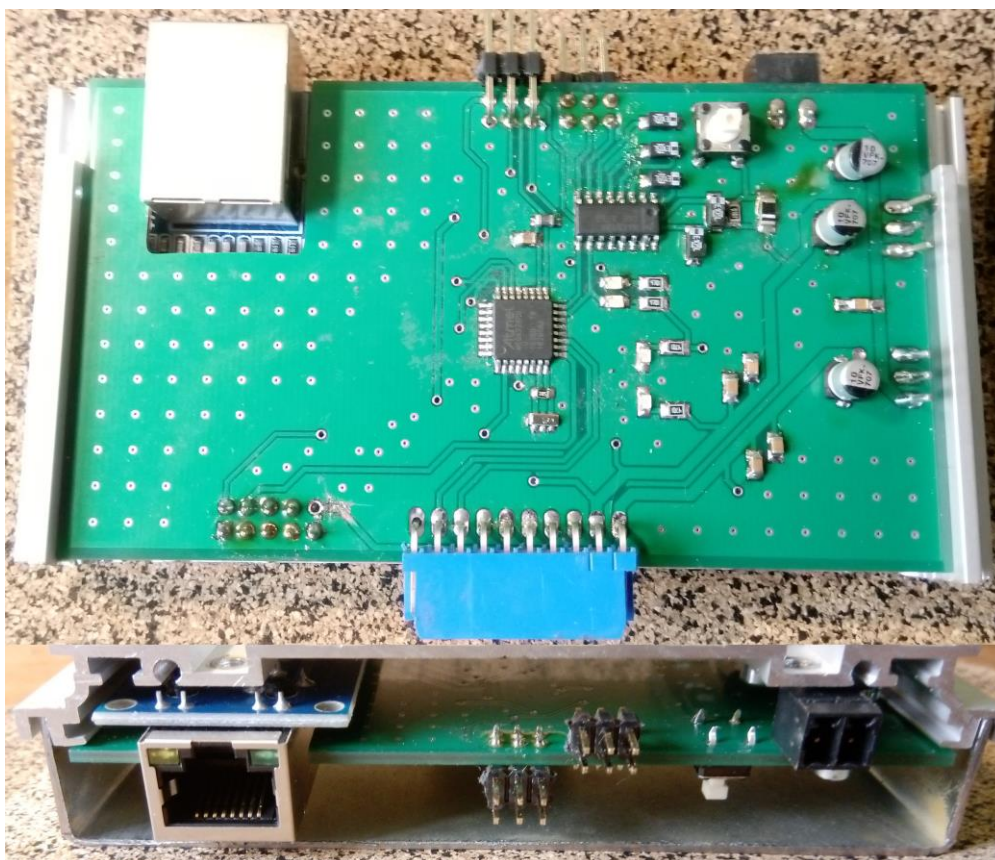
Obr. 40. prostředí programu Autodesk Fusion 360 s modelem jednotky osvětlení

3.5.2.5 Kompletace jednotek

Po vydrézování hliníkového profilu byl tento doplněn o spodní plechovou část šasi pro obě části zařízení.



Obr. 41. Kompletace jednotky světla s osazeným průhledovým plexi a DPS snímačů umístěnou na kontrolní diodě



Obr. 42. Kompletace řídicí jednotky

3.6 Návrh FW

Jako vývojové prostředí bylo použito AtmelStudio verze 7.0, zavádění programu do procesoru bylo realizováno pomocí USBASP programátoru. Program je napsaný v jazyce C pro konkrétní mikrokontrolér Atmega328.

3.6.1 Mapa registrů

V této kapitole je popsáno vnitřní rozložení registrů, práva k jejich čtení nebo zápisu. První část mapy registrů v rozsahu adres 0x00 až 0x64 je určena pro registry řídicí jednotky. Rozsah adres 0x65 až 0xC8 je určen pro informace ze jednotky světla. Rozsah adres 0xC9 až 0xFF je určen pro příkazy zařízení

Tab. 20. Mapa registrů řídicí část

Adresa	Práva	Velikost	Popis	
0	RW	4	IP adresa	[192][168][0][2]
4	RW	6	MAC adresa	[1][2][3][4][5][6]
A	RW	4	Brána	[192][168][0][1]
E	RW	4	Maska podsítě	[255][255][255][255]
12	RW	2	Port	502
14	RW	2	Cílový port	502
16	RW	4	Cílová IP adresa	[192][168][0][6]
1A	RO	1	Verze FW	1
1B	RO	2	Počet motominut	-
1D	RO	2	Počet motohodin	-
1F	RW	4	Nová IP adresa	-
23	RW	6	Nová MAC adresa	-
29	RW	4	Nová Brána	-
2D	RW	4	Nová maska podsítě	-
31	RW	4	Nová cílová adresa	-

Tab. 21. Mapa registrů části světla

Adresa	Práva	Velikost	Popis	
64	RW	1	R složka, intenzita zadaná	0
65	RW	1	G složka, intenzita zadaná	0
66	RW	1	B složka, intenzita zadaná	0
67	RW	1	W složka, intenzita zadaná	0
68	RO	1	R složka intenzita měřená na kontrolní diodě	-
69	RO	1	G složka intenzita měřená na kontrolní diodě	-
6A	RO	1	B složka intenzita měřená na kontrolní diodě	-
6B	RO	1	W složka intenzita měřená na kontrolní diodě	-
6C	RO	2	Hodnota intenzity osvětlení na kontrol. Diodě	-
6E	RO	1	Teplota světla, celočíselná část snímač 1	-
6F	RO	1	Teplota světla, desetinná část, snímač 1	-
70	RO	1	Teplota světla, celočíselná část snímač 2	-
71	RO	1	Teplota světla, desetinná část, snímač 2	

Tab. 22. Mapa příkazových registrů

Adresa	Práva	Velikost	Popis
C8	RW	1	Světlo ON (>0)/OFF(=0)

3.6.2 Práce s I2C sběrnici

Pro práci s I2C sběrnici bylo použito knihovny `<util/twi.h>`. Princip komunikace se snímači je popsán v následujícím příkladu komunikace se snímačem intenzity osvětlení BH1750.

```
unsigned char getDataBH(unsigned char adres, unsigned char *BHarr){
    adres = (adres<<1); //bitovy posun 7bitove adresy
    if(i2c_start(adres + I2C_WRITE)){ //start sekvence = 0 pri prijeti ACK
        i2c_stop(); // zastav sbernici pri chybe
        return 'h'; // chybova hlaska
    }
    else{
        i2c_start_wait(adres + I2C_WRITE); //start sekvence
        i2c_write(0x10); //zapis adresu registru
        if(i2c_rep_start(adres + I2C_READ)){ //pro slave - budeme cist
            i2c_stop(); //zastav pri chybe
            return 'g'; // chybova hlaska
        }
        else{
            BHarr[10] = i2c_readAck(); //precti horni byte
            BHarr[11] = i2c_readAck(); //precti dolni byte
            i2c_stop(); //zastav prenos
            return 'a'; //zprava korektne prijata
        }
    }
}
```

Z výše uvedeného zdrojového kódu je patrné, že při zavolání funkce `getDataBH` pro obsluhu snímače BH1750 dojde nejdříve k bitovému posunu 7-bitové adresy a doplnění bytu o bit indikující zápis. V dalším cyklu je očekáván ACK bit od slave zařízení. Pokud zařízení neodpoví, zastaví se provoz na sběrnici a funkce navrací chybovou hlášku „h“. Při přijetí ACK bitu následuje další start sekvence a odeslání adresy registru, který budeme číst – v tomto případě se jedná o 16-bitový registr obsahující hodnotu intenzity osvětlení v jednotkách lux). Dále je snímač informován o tom že v následujících cyklech budou požadována data z registru adresovaného v předchozí zprávě. Po přijetí ACK bitu následuje čtení 2 bytů, jenž na sběrnici odesílá snímač. Při korektním příjmu funkce vrací návratovou hodnotu „a“.

3.6.3 Práce s SPI sběrnici

Pro práci s SPI sběrnici při komunikaci byly vytvořeny pomocné funkce. Pro příklad je uveden funkce `W_common`, pomocí které je prováděn zápis do společného registru socketů kontroléru W5500.

Funkci s návratovou hodnotou typu `unsigned char` je předáván dolní byte adresy registru do kterého se bude zapisovat a hodnota pokynu, která se bude do registru zapisovat. Na počátku funkce je vytvořena proměnná, do níž bude po vykonání funkce odložena a následně navrácena odpověď kontroléru. Pomocí `CS_L` je zvolen slave se

kterým je komunikováno (v tomto případě je pouze jeden slave). V následujícím odstavci je popsáno odeslání jednoho rámce SPI zprávy:

Zápisem bytu do datového registru `SPDR1`, je iniciované odeslání zapsaného bytu na sběrnici SPI. Nyní je nutné počkat až do doby, než dojde k odeslání bytu – pomocí funkce `while` je kontrolován obsah registru `SPSR1`, konkrétně jeho bitu `SPIF1`. Pokud je `SPIF1` roven 1, znamená to, že zpráva byla úspěšně odeslána a je možné pokračovat v přenosu. Následuje přečtení datového registru `SPDR1`, čímž je provedeno vyprázdění registru a zároveň příprava pro odeslání dalších dat.

Výš popsaným způsobem jsou ve funkci odeslány celkem 4 datové byty, z čehož pouze po odeslání čtvrtého - očekáváme odpověď. Odpověď je následně uložena do proměnné, následuje „puštění“ slave zařízení pomocí `CS_H` a vrácení odpovědi pomocí návratové hodnoty funkce.

```
#define CS_L PORTD &= ~(1<<PORTD5)
#define CS_H PORTD |= (1<<PORTD5)

unsigned char W_common(unsigned char adresa, unsigned char pokyn){
    unsigned char odp = 0;           //proměnná pro odpověď
    CS_L;                             //Slave select pin = 0
    SPDR1 = 0x00;                     //vrchní byte adresy
    while(!(SPSR1 & (1<<SPIF1)));     //počkej na dokončení přenosu (flag
    bit v SPIF1)
    SPDR1;                             //přečtšný datového registru
    SPDR1 = adresa;                   //spodní byte adresy
    while(!(SPSR1 & (1<<SPIF1)));
    SPDR1;
    SPDR1 = 0x04;                     //zapisovaná hodnota do adresovaného
    registru
    while(!(SPSR1 & (1<<SPIF1)));
    SPDR1;
    SPDR1 = pokyn;
    while(!(SPSR1 & (1<<SPIF1)));
    odp = SPDR1;                       //přečtení datového registru
    (očekávaná odpověď)
    CS_H;                             //Slave select pin = 1
    return odp;                       //návratové hodnota funkce
}
```

Další definované funkce pro komunikaci a práci s ethernet kontrolérem, respektive Socketem 0.

```
unsigned char W_Socket_0(unsigned char adresa, unsigned char pokyn);
unsigned char R_Socket_0(unsigned char adresa, unsigned char pokyn);
unsigned char W_TX_Socket_0(unsigned char adresa, unsigned char pokyn);
unsigned char R_TX_Socket_0(unsigned char adresa, unsigned char pokyn);
unsigned char R_RX_Socket_0(unsigned char adresa, unsigned char pokyn);
unsigned char W_RX_Socket_0(unsigned char adresa, unsigned char pokyn);
```


3.6.4 Zpracování MODBUS

Zpracování příchozí zprávy MODBUS je odlišné pro příjem po sériové lince a pro příjem přes ethernet. V následujících příkladech jsou popsány oba dva případy.

3.6.4.1 Zpráva MODBUS po seriové lince

Při příjmu zprávy je tuto nutné zpracovat v přerušení od seriové linky. Pokud dojde k příjmu znaku na seriové lince je zavoláno následující přerušení:

```
ISR(USART1_RX_vect)
{
    unsigned char RCB; // promenna pro ulozeni prijateho bytu
    RCB = UDR1 ;        // vycitni znaku z bufferu
    if(zprava == 0){    // mame prazdno?
        if(cnt == 0){  // jedna se o prvni znak zpravy?
            if (RCB == SLAVE_ID){ //je zprava pro nas?
                pole[cnt]=RCB;      //uloz
                cnt = 1;            //inc counter znaku
            }
        }
        else{          // dalsi znaky zpravy
            pole[cnt]=RCB; //uloz
            cnt = cnt + 1; //inc counter
            if (cnt == 4){ //hodnota fce modbus
                if(pole[1]==0x03){ //fce 03
                    buffer = (pole[2]*2); //pocitej velikost
                    //zpravy
                    buffer = buffer + 6; //+ rezije
                }
                else if(pole[1]==0x06){ //fce 06
                    //bytu
                    buffer = 8 ; //velikost zpravy 8
                }
            }
            else if (cnt == 7){ //az bude prijato minimalne 7
                //bytu
                if(pole[1]==0x10){ //fce 16
                    //zap bytu
                    buffer = pole[5]*2; //velikost dle poctu
                    buffer = buffer + 9; // plus rezije
                }
            }
            if (cnt == buffer){ //konec zpravy
                cnt = 0;
                zprava = 1; //zprava prijata
            }
        }
        T3_5 = 0; // nuluj timer
    }
}
```

Z výše uvedeného zdrojového kódu je patrné, že zpráva se načítá po jednotlivých bytech ze zásobníku přijímače seriové linky. Při příjmu zprávy je očekáván správný tvar zprávy MODBUS a dle toho je příjem jednotlivých bytů kontrolován. Důležitou akcí je

zde nulování proměnné T3_5 vždy po přijetí znaku (provedení přerušení). Hodnota proměnné se upravuje a kontroluje v cyklu časovače, kde se v časových intervalech jedné milisekundy inkrementuje a pokud dosáhne hodnoty 4, znamená to, že zpráva je již přijata, nebo došla nekompletní (vyhodnoceno jinde v programu) – v obou případech ale dojde k nulování počítadla přijatých bytů a nově příchozí znak iniciuje nové načítání zprávy.

Přijatá zpráva je pak uložena v poli znaků a pomocí proměnné `zprava` je při pokračování v běhu programu program informován a zpráva je následně zpracovávána.

Odesílání zprávy ze zařízení je realizováno přípravou zprávy do vyhrazeného paměťového prostoru v mikropočítači a pomocí „for“ cyklu jsou následně jednotlivé byty sestavené zprávy vyslány na seriovou linku.

3.6.4.2 Zpráva MODBUS po ethernetu

Princip příjmu zprávy po ethernetu je odlišný z důvodu použitého ethernet kontroleru. Kontroler je vybavený vnitřní pamětí 32kB, do které ukládá jednotlivé příchozí zprávy ze sítě. Při běhu programu je pak pravidelně kontrolován registr `S0_IR`, jehož chování je v kontroléru nastaveno tak, že pokud přijme zprávu, nastaví hodnota bitu přiřazeného danému socketu do 1. Pokud je tedy při vyčítání obsahu registru zjištěno, že hodnota registru není nulová, provede se v programu procesoru příjem zprávy z ethernet kontroléru a ta se následně zpracovává.

Princip vyčítání z ethernet kontroléru je následující:

- Cyklická kontrola obsahu registru přerušení `S0_IR`, pokud je nenulový, následuje:
- Vyčtení registru `S0_RX_RD` obsahujícího adresu počátku zprávy v paměti ethernet kontroléru
- Vyčtení registru `S0_RX_RSR` obsahujícího velikost přijaté zprávy v paměti ethernet kontroléru
- Vyčtení počtu registrů od adresy počátku zprávy v počtu určeném velikostí zprávy a jejich uložení do pole bytů
- Zapsání poslední vyčítané adresy do registru `S0_RX_RD` obsahujícího adresu počátku zprávy (od tohoto místa pak v ethernet kontroléru proběhne uložení další zprávy)
- Smazání hodnoty registru přerušení `S0_IR`
- Zaslání příkazu 0x40 do socketového příkazového registru (informace pro ethernet kontrolér o vyzvednutí zprávy)

Přijatá zpráva je pak následně zpracovaná v běhu programu.

Princip odeslání zprávy přes ethernet kontrolér je následující:

- Vyčtení aktuální hodnoty registru Socket Write Pointer (S0_TX_WR), který obsahuje ukazatel na konec poslední zapsané hodnoty v zásobníku pro odeslání
- Odeslání dat do zásobníku na pozici určenou S0_TX_WR
- Součet hodnoty registru S0_TX_WR s počtem odesílaných bajtů v zásobníku
- Zápis nové hodnoty S0_TX_WR
- Zaslání příkazu 0x20 do socketového příkazového registru, Ethernet kontrolér následně odešle zprávu o velikosti dané předchozím zápisem do registru a je připraven na příjem dalších dat z mikrokontroléru k odeslání.

3.6.5 Ovládání inteligentních LED diod

Pro řízení a komunikaci s inteligentními LED byla použita a upravena volně dostupná knihovna [10]. Z knihovny byla využita a upravena funkce `led_strip_write` a definice datového typu `typedef struct rgb_color`. Výše uvedená knihovna byla upravena tak, aby ji bylo možné použít pro inteligentní LED diody se 4 baravami (knihovna je původně napsaná pouze pro RGB verzi dioda, úprava je funkční s RGBW diodami), s mikropočítačem Atmega328PB a externím oscilátorem 16MHz.

```
typedef struct rgb_color
{
    unsigned char red, green, blue, white;
} rgb_color;
rgb_color colors[LED_COUNT];
```

Funkce pro ovládání inteligentních LED je naprogramovaná v jazyce strojových instrukcí ASM.

```

void led_strip_write(rgb_color * colors, unsigned int count)
{
    // Set the pin to be an output driving low.
    LED_STRIP_PORT &= ~(1<<LED_STRIP_PIN);
    LED_STRIP_DDR |= (1<<LED_STRIP_PIN);
    cli(); // Disable interrupts temporarily because we don't want our pulse
    timing to be messed up.
    while(count--)
    {
        // Send a color to the LED strip.
        // The assembly below also increments the 'colors' pointer,
        // it will be pointing to the next color at the end of this loop.
        asm volatile(
            "ld __tmp_reg__, %a0+\n"
            "ld __tmp_reg__, %a0\n"
            "rcall send_led_strip_byte%=\n" // Send red component.
            "ld __tmp_reg__, -%a0\n"
            "rcall send_led_strip_byte%=\n" // Send green component.
            "ld __tmp_reg__, %a0+\n"
            "ld __tmp_reg__, %a0+\n"
            "ld __tmp_reg__, %a0+\n"
            "rcall send_led_strip_byte%=\n" // Send blue component.
            "ld __tmp_reg__, %a0+\n"
            "rcall send_led_strip_byte%=\n" // Send White component
            "rjmp led_strip_asm_end%=\n" // Jump past the assembly subroutines.
        // send_led_strip_byte subroutine: Sends a byte to the LED strip.
        "send_led_strip_byte%=: \n"
        "rcall send_led_strip_bit%=\n" // Send most-significant bit (7).
        "rcall send_led_strip_bit%=\n"
        "rcall send_led_strip_bit%=\n"
        "rcall send_led_strip_bit%=\n"
        "rcall send_led_strip_bit%=\n"
        "rcall send_led_strip_bit%=\n"
        "rcall send_led_strip_bit%=\n"
        "rcall send_led_strip_bit%=\n" // Send least-significant bit (bit 0).
        "ret\n"
        "send_led_strip_bit%=: \n"
        "sbi %2, %3\n" // Drive the line high.
        "rol __tmp_reg__\n" // Rotate left through carry.
        "nop\n" "nop\n"
        "brcs .+2\n" "cbi %2, %3\n"// If the bit to send is 0, drive the line
low now.
        "nop\n" "nop\n" "nop\n" "nop\n" "nop\n"
        "brcc .+2\n" "cbi %2, %3\n"// If the bit to send is 1, drive the line
low now.
        "ret\n"
        "led_strip_asm_end%=: "
        : "=b" (colors)
        : "0" (colors), // %a0 points to the next color to display
        "I" (_SFR_IO_ADDR(LED_STRIP_PORT)), // %2 is the port register (e.g.
PORTC)
        "I" (LED_STRIP_PIN) // %3 is the pin number (0-8)
        );
    }
    sei(); // Re-enable interrupts now that we are done.
    _delay_us(80); // Send the reset signal.
}

```

4. ZÁVĚR

Ve své bakalářské práci jsem se zabýval vývojem a realizací multispektrálního osvětlení pro průmyslové vidění.

V teoretické části jsem nastudoval problematiku strojového a průmyslového vidění, jednotlivé používané typy osvětlení a vliv použitého osvětlení na výslednou kvalitu snímku.

V hlavní části práce postupně popisuji vznik zařízení rozdělený do tematických celků. V návrhu zařízení se zabývám použitými komunikačními protokoly a jejich implementací pro konkrétní řešení a mikropočítač. Použitím vybraných snímačů neelektrických veličin, jejich následným převodem a zpřístupněním nadřazenému systému.

Pro komunikaci se zařízením byl zvolen protokol Modbus TCP/IP po ethernetu a Modbus RTU po seriové lince a to z důvodu používání protokolů v průmyslových aplikacích, jejich historii a dobré dostupnosti studijních materiálů.

Z pohledu dalšího vývoje projektu by bylo vhodné umožnit pohodlnější práci s celým osvětlením pomocí vývoje počítačového grafického rozhraní pro ovládání světla. Dalším vhodným rozšířením by bylo vytvoření více druhů jednotek osvětlení pro výzkumné účely. Vzhledem k modulární koncepci zařízení lze světelných jednotek vytvořit neomezené množství typů a to při dodržení zapojení spojovacího konektoru s řídicí jednotkou.

Literatura

- [1] A Practical Guide to Machine Vision Lighting [online]. Daryl Martin, 2017 [cit. 2018-11-05]. Dostupné z <http://www.ni.com/white-paper/6901/en/#toc2>
- [2] Strojové vidění – principy a charakteristiky [online]. Otto Havle, 2008 [cit. 2018-11-06]. Dostupné z http://www.automa.cz/cz/casopis-clanky/strojove-videni-i-principy-a-charakteristiky-2008_01_36550_5518/
- [3] Datasheet BH1750 [online]. 2012[cit. 2018-12-20]. Dostupné z http://cpre.kmutnb.ac.th/esl/learning/bh1750-light-sensor/bh1750fvi-e_datasheet.pdf
- [4] Datasheet TCS3472 [online]. 2012[cit. 2018-12-20]. Dostupné z <https://cdn-shop.adafruit.com/datasheets/TCS34725.pdf>
- [5] Datasheet BMP280 [online]. 2018[cit. 2018-12-20]. Dostupné z https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BMP280-DS001.pdf
- [6] I2C basics [online]. Mayank (Max), 2014[cit. 2018-12-20]. Dostupné z https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BMP280-DS001.pdf
- [7] GADRE, Dhananjay V. *Programming and customizing the AVR microcontroller*. New York: McGraw-Hill, c2001. ISBN 00-713-4666-X.
- [8] MANN, Burkhard. C pro mikrokontroléry: ANSI-C, kompilátory C, spojovací programy - linkery, práce s ATMEL AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky ... Praha: BEN - technická literatura, 2003. ISBN 80-730-0077-6.
- [9] KABELOVÁ, Alena a Libor DOSTÁLEK. Velký průvodce protokoly TCP/IP a systémem DNS: ANSI-C, kompilátory C, spojovací programy - linkery, práce s ATMEL AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky ... 3. aktualiz. a rozš. vyd. Praha: Computer Press, 2002. Všechny cesty k informacím. ISBN 80-722-6675-6.
- [10] Example AVR code for addressable RGB LED strips from Pololu [online]. ©Pololu.com, 2016, [cit. 2019-05-01]. Dostupné z: <https://github.com/pololu/pololu-led-strip-avr>
- [11] HOROWITZ, Paul. *The art of electronics*. Third edition. New York, NY: Cambridge University Press, [2015]. ISBN 978-0-521-80926-9.
- [12] Sériová rozhraní [online]. Dudáček K., 2002 [cit. 2019-05-01]. Dostupné z: http://home.zcu.cz/~dudacek/NMS/Seriova_rozhrani.pdf
- [13] SPI Block Guide [online]. Motorola Inc. 2000 [cit. 2019-05-01]. Dostupné z: <https://www.datasheetarchive.com/pdf/download.php?id=57a1b96e07f0b22cf754bb03ce514d5548b83a&type=M&term=S12SPIV3>
- [14] Datasheet Atmega328PB [online]. ,2018 [cit. 2019-05-01]. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/40001906C.pdf>

Seznam příloh

- Příloha 1 – Zdrojový kód programu je uložen na přiloženém CD
- Příloha 2 – 3D model jednotky osvětlení je uložen na přiloženém CD
- Příloha 3 – 3D model řídicí jednotky je uložen na přiloženém CD
- Příloha 4 – Projekt DPS řídicí jednotky je uložen na přiloženém CD
- Příloha 5 – Projekt DPS jednotky osvětlení je uložen na přiloženém CD
- Příloha 6 – Projekt DPS snímačů je uložen na přiloženém CD
- Příloha 7 – Datasheet mikropočítače Atmega328PB je uložen na přiloženém CD
- Příloha 8 – Datasheet snímače TCS34725 je uložen na přiloženém CD
- Příloha 8 – Datasheet snímače BH1750 je uložen na přiloženém CD
- Příloha 8 – Datasheet snímače MCP9834 je uložen na přiloženém CD